Textual Bot for Mobile Application with MySQL, Python, and GPT-4

A PROJECT REPORT

Submitted by,

Pawan Kumar D - 20211CSD0087
Nihal Gagan Kunte - 20211CSD0079
Sai Abhishek - 20211CSD0075
Darshan Naik - 20211CSD0061
S Punith Kumar - 20211CSD0159

Under the guidance of,
Dr. S Saravana Kumar
Associate Professor

in partial fulfillment for the award of the degree of
BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING
[DATA SCIENCE]
At

PRESIDENCY UNIVERSITY
BENGALURU
DECEMBER 2024

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **"**Textual Bot Mobile Application Using MySQL, Python, and GPT-4**"** being submitted by "Pawan Kumar D , Nihal Gagan Kunte , Sai Abhishek ,Darshan Naik,S Punith Kumar " bearing roll number(s) "20211CSD0087,20211CSD0079,20211CSD0075, 20211CSD0061, 20211CSD0159" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING[DATA SCIENCE] is a bonafide work carried out under my supervision.

Dr. S Saravana Kumar
Associate Professor
Presidency School of IS
Presidency University

Dr. Saira Bhanu
Professor & HoD
Presidency School of CSE
Presidency University

Dr. L. SHAKKEERA
Associate Dean
Presidency School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
Presidency School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean – Presidency School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Textual Bot for Mobile Application with MySQL, Python, and GPT-4** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science Engineering[Data Science]** , is a record of our own investigations carried under the guidance of **Dr. S Saravana Kumar , Associate Professor, School of Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Pawan Kumar D - 20211CSD0087
Nihal Gagan Kunte - 20211CSD0079
Sai Abhishek - 20211CSD0075
Darshan Naik - 20211CSD0061
S Punith Kumar - 20211CSD0159

ABSTRACT

The fast-paced evolution in conversational AI has reimagined chatbots into dynamic, smart tools that hold intelligent conversations. This work will attempt to outline the development of a mobile chatbot application with GPT-4, Python, and MySQL for applications as varied as customer support, personal assistance, and knowledge sharing. In this work, the developed chatbot utilizes GPT-4 for advanced natural language processing to generate responses to any situation, mimicking human-like dialogue, believed to further enhance user experience and engagement.

Python will be forming the backbone for the backend logic, making the functionality strong, and MySQL for structured and secure data storage. On top of this, the system will also have personalized recommendations, secure authentication, and real-time responsiveness to the queries that the user asks. Optimized system design and implementation definitely resolve the key challenges associated with latency, scalability, and data security.

This project encompasses the integration of cutting-edge AI with scalable backend technologies and thus presents an intelligent and efficient chatbot solution that is also amiable in nature. Such versatility will be appropriate for multiple domains because it ensures high-quality interaction in improving overall user satisfaction.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, Presidency School of Engineering and Dean, Presidency School of Computer Science And Engineering & Information Science, Presidency University for getting us permission to undergo the project.We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** Presidency School of Computer Science And Engineering, Presidency University, and "**Dr. Saira Bhanu**", Head of the Department, Presidency School of Computer Science And Engineering, Presidency University, for rendering timely help in completing this project successfully.We are greatly indebted to our guide **Dr. S Saravana Kumar Associate Professor** , Presidency School of Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators "**Dr. Afroz Pasha**" and Git hub coordinator **Mr. Muthuraj.**

Pawan Kumar D - 20211CSD0087

Nihal Gagan Kunte - 20211CSD0079

Sai Abhishek - 20211CSD0075

Darshan Naik - 20211CSD0061

S Punith Kumar - 20211CSD0159

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 Background

The digital age has irrecoverably seen a rise in conversational AI technologies, which completely revolutionized how businesses communicate with their customers and people seek information. Textual bots, important subcategories of conversational AI, have evolved from simple question-and-answer systems to sophisticated virtual assistants capable of subtle, contextually aware conversations. This evolution is largely attributed to the advancement of natural language processing and the availability of powerful language models like GPT-4.

From e-commerce and healthcare to education and further entertainment, the list of industries using chatbots keeps on increasing. They upgrade the customer experience by offering instant support round the clock while saving businesses operational costs. This involves the integration of robust backend systems, like MySQL databases, that make such applications efficient in handling big data at a scale of millions of interactions every day.

1.2 Problem Statement

Despite the proliferation of Chatbots, there is a number of challenges that still remain:

Limited Contextual Understanding: Most of the bots cannot keep the context for a long time, which results in repetition or irrelevant responses.

Data Management Issues: Safe storing and retrieval of user data remain one of the major challenges, especially in applications where personalization is needed.

Customization and scalability: Most chatbots are built for specific applications and therefore can't cater to a diverse range of requirements from different domains and users.

Integration Complexity: Advanced AI models require technical experience in integrating them with their respective infrastructures, which involves a great deal of computational resource usage and months of development.

User Trust and Privacy: A lot of users don't use bots because of not being open and not having faith in how their data is used.

1.3 Objectives

The main aims of this project will be:

1. To design a multi-purpose textual chatbot that is contextually aware of previously discussed topics in a conversation.

2. To implement an efficient, MySQL-based database that can achieve scalability and security in data storage and retrieval.

3. It will be applied to GPT-4 for developing advanced conversational capabilities, considering nuanced and empathetic responses.

4. To design a user-friendly mobile application that guarantees smooth interaction due to intuitive navigation.

5. To evaluate the chatbot's performance in real-world scenarios across various domains and identify areas for improvement.

6. To comply with data privacy regulations by providing users with control over their data.

1.4 Project Scope

This project aims at developing a robust and flexible chatbot system for use in several domains. For this phase, it has targeted customer service, but the architecture shall be open to extension for use in healthcare, education, and e-commerce. Integration with MySQL into the system is crucial to make sure scalability works efficiently with an increasing amount of users or demands. The major language used in the system is Python, supplemented with an extensive library and compatibility with AI frameworks. In addition, the mobile application will ensure that there is platform independence, and therefore it can be accessed with Android and iOS devices..

CHAPTER-2
LITERATURE SURVEY

1. AI-based FAQ chatting bots

Overview: It has designed these chatbots to increase user interaction-voice assistant improved interactions, AI-powered responses in real-time. Advanced integration of NLP allows them to understand the query of the user more precisely.

Example: A concept was put forward for an AI-driven voice assistant FAQ chatbot, which would provide an enhanced user experience through more personalized and contextual interactions

Challenges: Limited interactivity in traditional systems and accessibility problems for disabled users.

Improvements: Voice recognition and machine learning make for even more intuitive and versatile user interactions.

2. Chatbots in Educational Institutions

Overview: In academic institutions, chatbots help in the dissemination of information, solving queries, and counseling. They answer all admission-related queries and also cater to course-related information.

Key Findings: Chatbots use neural networks, sequential modeling, and AIML for accurate and context-aware interactions

They improve accessibility by being available any time and reduce dependency on human intervention.

Example: An institutional chatbot integrates voice and text inputs, proving to reduce the manual workload, and provides an immersive experience for users

3. Chatbots for Digital Business Transformation

Overview: In simple terms, chatbots play a very crucial role in the process of digital transformation through automated customer service, CRM improvement, and the facilitation of day-and-night customer support. The conversational agent will help the company to get more engaged with users and assure loyalty.

Key Contributions: Address the role of chatbots in industries such as retail, health care, and finance. Highlight the shift towards AI-driven systems for customer interaction.

Challenges: Privacy, complex queries, and enhancement of user satisfaction.

Future Directions: Emphasis on Interdisciplinary Research, and Real-time User Feedback Mechanisms

4. Chatbots for Mobile Applications

Overview: Mobile chatbots are designed to meet the expectations of users for smooth interactions. It supports both structured and unstructured queries in real time.

Example: Dalmia Cement's mobile chatbot provides real-time order updates, improving customer engagement

Technologies Used: Cross-platform solutions like Flutter and Dart.

Chatbot engines designed to understand industry-specific terminology.

Challenges: Security vulnerabilities and the development of reliable state management solutions.

5. Historical and Technological Development of Chatbots

Overview: From the very early days of ELIZA to modern AI-powered systems, chatbots have evolved significantly**.**

Key Milestones: ELIZA employed keyword matching techniques but with no sense of context. Modern systems use deep learning such as LSTM, and semantic analysis for better performance.

Use Cases: Some of the different applications that are currently supported by chatbots include health, education, and customer support.

6. System Design and Implementation

Proposed Architectures: Web services based on SOAP can be used along with voice recognition modules for better response times.

Customizable databases and self-learning modules for improved adaptability and user satisfaction

Examples: Chatbots Using AIML for Dynamic and Context-Sensitive Interaction

7. Challenges and Future Directions

Challenges: Limited database content leads to unsatisfactory responses.

Data protection and privacy concerns within sensitive environments.

Need for better optimization to handle complex queries

Future Directions: Emotional intelligence with advanced sentiment analysis is integrated.

Expansion of knowledge bases to widen their application scope.

Real-time learning and adaptability to the needs of users in continuous evolution.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

Although there are some very active developments in chatbot technologies, a few gaps remain in the existing methods that this project tries to fill in:

3.1 Limited Contextual Awareness

Most of the chatbots don't maintain context over a prolonged period of conversation. This leads to frustration and then disengagement from the user. Most of these systems are either dependent on fixed scripts or suffer from limited memory, which ultimately makes them weak in understanding complex or evolving queries.

3.2 Lack of Personalization

Current bots do not have the potential to personalize responses in accordance with user preferences and histories.There is a lack of strong user data integration, which limits the ability to dynamically suggest information or services.

3.3 Security and Privacy Challenges

Data privacy is one factor. Most of the chatbots do not follow global standards such as GDPR.Poor mechanisms for encryption and poor authentication can expose potential breaches to information sensitive to end-users.

3.4 Scalability Restrictions

Most chatbots have several performance issues when it has to handle higher traffic or large-scale implementation.Limited database scalability hinders the ability to manage growing user bases and data volumes effectively.

3.5 Limited Multilingual Support

Most chatbots provide support for a limited number of languages, making them inaccessible to diverse user groups.Poor handling of regional dialects or mixed language queries reduces usability in global contexts.

3.6 Bad Integration with Advanced AI Models

Most organizations have found integrating a high-level model such as GPT-4 an extremely cumbersome task, resource consuming.Limited support for integrating AI-driven conversational capabilities with traditional database-driven systems.

3.7 Unable to Recognize Emotion and Sentiment

Most of the existing systems do not recognize and respond to user emotions or sentiments. This gap leads to impersonal interactions, especially in domains such as mental health support or customer care.

3.8 Inflexible Domain Adaptation

Currently, much retraining and configuration is required to adapt the chatbots for new domains or industries.

Lack of modularity in design limits reusability and usage across a wide range of applications.

Addressing these research gaps, this project will lead to a next-generation chatbot solution that features state-of-the-art AI capabilities, solid data management, and a user-oriented design.

CHAPTER-4
OBJECTIVES

The objectives of the chatbot application have been set in such a way that the solution will be all-inclusive, user-centered, and sound technically. The objectives range from functional goals, technical requirements to long-term sustainability.

Functional Objectives

Provide Seamless User Interaction

Provide intuitive, human-like conversations using the state-of-the-art natural language processing powers of GPT-4.

Support a wide variety of use cases, from customer service to education and e-commerce support.

Improve Accessibility and Usability

This work shall focus on developing a cross-platform, mobile-friendly application for users with various levels of technical proficiency.

Features can be added, such as multi-language support and voice-to-text capability for inclusivity.

Personalization

Store user preferences and interaction history for use in providing personalized responses and recommendations.

This allows users to adjust the behavior and tone of chatbots to their liking.

Real-Time Responsiveness

The Technical Objectives

Scalable Architecture of a System

Design a system that can support high user traffic without affecting performance or availability.

Use cloud-based hosting like AWS, Google Cloud, etc., which will allow for dynamic scalability.

Robust Data Management

Design an optimized relational database (MySQL) to store the user profiles, preferences, and conversation logs.

Ensure consistency of data, fast retrieval, and storage in a secure way.

State-of-the-Art AI Integration

Integrate GPT-4 for more sophisticated conversational AI applications, fine-tuned to the application at hand.

Continuously improve the AI through supervised learning and user feedback.

Secure Application Development

Encrypt sensitive data with end-to-end encryption to prevent unauthorized access.

Use OAuth 2.0 and RBAC for secure authentication protocols that ensure the security of user data.

Testing and Validation Objectives

Ensure Quality and Reliability

Do intense unit and integration testing, system testing to detect and fix bugs or vulnerabilities.

Automate using Selenium for the frontend, Postman for API validation.

Collect User Feedback

Perform beta testing with a diversified group in order to gather insight into usability and functionality.

Refine the chatbot system based on iterative feedback.

Validate Performance Metrics

Response times, user engagement, and the accuracy of chatbot responses should be monitored to ensure operational efficiency.

Objectives of Deployment and Maintenance

Efficient Deployment

Deploy the chatbot on a cloud infrastructure that can scale with the least amount of downtime.
Ensure that all components at the deployment stage are smoothly fitted together.
Continuous Monitoring and Updates

Use analytics tools to monitor system performance, user satisfaction, and potential issues.
Roll out frequent updates that take care of bugs, functionality issues, and fit in new features.
Adaptability to Evolving Needs

Be able to evolve the chatbot to changing requirements, for example, due to new use cases or changes in regulation.
Enable modular system updates to reduce redevelopment efforts.

Compliances and Ethical Objectives

Follow Data Privacy Regulations

Make sure that the application is GDPR, CCPA, and other data protection law compliant.

Provide mechanisms for user consent and transparent data handling policies.

Ensure Ethical AI Use

Avoid biases in chatbot responses by testing AI behavior on diverse data sets. Provide users with disclaimers or transparency about AI usage. Promote Confidence and Transparency Clearly state how users' data will be used, stored, and protected. Give users the opportunity to view, edit, and delete their data at any time. 6.

Business and Market Objectives Improve Operational Efficiency Automate repetitive queries and tasks that require less human intervention and more productivity. Offer businesses valuable insights for action by analytics of user behavior and preference. Increase User Engagement Design an interactive, engaging chatbot that will make users want to come back for more. Apply gamification aspects, like rewarding frequent users, to increase retention. Drive Revenue Growth Enable monetization by either offering premium features, advanced AI assistance, or through integrations with third-party services. Assist businesses in lead generation and customer acquisition by providing insightful recommendations.

CHAPTER-5

PROPOSED METHODOLOGY

5.1 Overview

This is proposed, the methodology would allow for the systematic design, development, and deployment of a scalable and secure chatbot application. This methodology makes the most of leading-edge technologies for conversational AI-GPT-4-and MySQL for data management to make the chatbot not only effective but also user-oriented.

The process is designed to always keep scalability and data security foremost, along with support for modern technology integrations-the chatbot would grow and mature with user requirements and system demand. This follows best practices across every phase with a view towards robust performance and adherence to industry standards.

5.2 Key Phases

5.2.1 Requirements Analysis

User Requirements: Engage the target users through questionnaires, interviews, and focus groups to understand their needs, behaviors, and expectations. It will inform functionality, tone, and use cases for the chatbot.

Technical Requirements: Identify and analyze all the essential tools, frameworks, APIs, and technologies that would be required in the development and maintenance of the system. This shall include GPT-4 API setup, Cloud Infrastructure, and Third-Party Services.

5.2.2 System Architecture Design

Frontend Design: It's designed to be mobile-first, and by using any framework like Flutter or React Native, it can be made sure that a chatbot works perfectly on all devices. The design will be simple and intuitive; this UX is intended to reduce user friction while increasing user engagement.

Backend Architecture: The backend shall be implemented in Python, utilizing either Flask or Django. It will be responsible for handling user requests, processing data, and communicating with GPT-4 for real-time conversations. The backend should be modularized to make scaling up easier and for maintenance purposes.

Database Design: This includes the design of a relational database schema that would be optimized with the use of MySQL. This would include the design of schemas or tables to record user profiles, their preference lists, history of session trails, and even statistics regarding chatbot performance. The application will provide indexing and optimize queries in a way to realize fast data retrieval and thus improve overall system performance.

5.2.3 Development

Chatbot Integration: GPT-4 will be integrated with the backend, facilitating conversational AI. It shall be fine-tuned for specific use cases: customer support, e-commerce assistance, or educational guidance. It will also be able to learn from ongoing interactions and improve over time.

Database Management: Apply CRUD - Create, Read, Update, Delete-operations in such a way that user data will be maintained securely and efficiently. Guarantee data integrity, consistency, and performance optimization by applying different advanced techniques: indexing, caching, partitioning.

Security Features: Include high-level security features such as end-to-end encryption of sensitive data, the use of OAuth 2.0 to authenticate and lock down API endpoints, and RBAC, ensuring users only have access to what they should have.

5.2.4 Testing and Validation

Unit Testing: Each separate module should be unit-tested extensively to ensure that it works. For each unit, tests should be developed to ensure expected behavior regarding responses by the bot, database interactions, and security mechanisms.

Integration Testing: This will test the integration of the frontend, backend, and GPT-4 API to ensure that all system components communicate well and the overall user experience is not disrupted.

User Testing: The system is to be beta tested on a wide variety of users to capture firsthand information on its usability, user experience, and performance. User testing insights will be used to refine the system and iron out any issues before the final release.

5.2 Deployment and Maintenance Cloud Deployment:

The home for the application of the chatbot will be some scalable cloud platforms like AWS, Google Cloud, or Microsoft Azure. The cloud infrastructure will grant the system dynamic scaling in regard to traffic and demand for resources, thus providing high availability and reliability.

Analytics and Monitoring Tools: Provide analytics and monitoring tools such as Google Analytics, New Relic, or Datadog to monitor key performance metrics on response times, user engagement, and error rates. This is meant to enhance the continuous observation of the health of the system and how users interact with it.

Regular Updates: Establish a plan for releasing periodic updates based on user feedback, bug fixes, and changing system requirements. The updates will be deployed in phases to minimize system downtime and ensure smooth transitions.

5.3 Tools and Technologies

Programming Language: Python will be used for the development of the backend because it is versatile, easy to integrate, and has a large ecosystem of libraries.

Frontend Frameworks: Flutter or React Native will be used for a cross-platform mobile application so that the application provides the same experience on both iOS and Android devices.

Database: MySQL shall be used for the storage of structured data, as it is reliable, easy to integrate, and does a great job at querying performance. Indexing and optimization will play an important role in handling big datasets.

It will integrate a state-of-the-art conversational AI model, such as GPT-4, for natural language understanding and generation. The model will undergo fine-tuning on specific use cases and will continue to improve through user interactions.

Security Tools: PyCrypto for encryption, OAuth 2.0 for secure user authentication. This will ensure that the system meets the highest standards of security.

Testing Tools: Test Automation Framework: Selenium for automated frontend testing and Postman for API testing. Both these tools will provide strong support for the testing framework to ensure system reliability and performance. 5.4 Flow Diagram The workflow diagram will represent the interaction of key components: User Input: The user interacts with the application's interface, which is the frontend, sending in queries or requests. Backend Processing: The request is further passed on from the frontend to the backend for processing using Python, such as Flask or Django. The back-end manages all the user data internally and interfaces with the GPT-4 API for the chatbot responses. AI Integration: The back-end makes API calls to GPT-4, which then generates the responses that should fit the context with respect to whatever query the user meant. Data Management: This information, which relies on user profiles in the MySQL database, should be fetched at the backend and responded to accordingly..

5.4 Detailed Implementation Timeline

Accordingly, the following is the proposed timeline to ensure smooth completion of this project, dividing it into phases with estimated durations for each:

Phase 1: Requirements Analysis 2-3 weeks
Surveying, interviewing, collecting user feedback: 1 week
Analyze technical and compliance requirements (1 week)
Develop system specifications and obtain stakeholder sign-off (1 week)

Phase 2: System Architecture Design, 3-4 weeks

Finalize frontend design and user interface: 2 weeks

Arch, design the backend components; integration with GPT-4 API would be done. (2 weeks)

Database schema design-review with optimization by 1 week

Phase 3: Development 8-10 weeks

Begin Python with backend development - Flask/Django, 3 weeks

Develop frontend using Flutter or React Native (3 weeks)

Integrate GPT-4 with the system-chatbot functionality on your own. Approximately: 2 weeks

Implement database CRUD operations and set up secure APIs (2 weeks)

Testing and Validation

Phase 4: 4-6 weeks

Unit testing of individual modules: 1 week

Integration testing with front-end, back-end, and GPT-4: 2 weeks

Conducting user testing with feedback gathering and bug fixing (1-2 weeks)

Phase 5: Deployment and Maintenance, 3-4 weeks

Deploy the application on cloud services (1 week)

Establish performance monitoring tools and analytics: 1 week

Plan and schedule periodic software updates - every 1-2 weeks.

This timeline allows for flexibility but keeps the direction clear toward successful project

5.5 Scalability Considerations

It is required that the application should be scalable to handle increased traffic as the user base grows without compromising on performance. The following strategies shall be implemented:

Load Balancing: Utilize load balancers to distribute the traffic of user requests across multiple servers so that no single server is overloaded.

Horizontal Scaling: Deploy parallel instances of the backend applications on each cloud platform, including AWS, Google Cloud; this will scale horizontally and assures extra resources to get added with the increased traffic coming your way.

Database Sharding: Database splitting into small pieces called shards, either based on user data or by geographical location. The rationale behind it is to make it more manageable for querying performance.

Caching: Implement mechanisms for caching, such as Redis, for frequently accessed data to reduce the load on the database.

Microservices Architecture: The backend application shall be further divided into smaller microservices if needed for future expansion, which can be independently scaled and maintained.

These strategies will help keep the system performant and responsive under variable loads.

5.6 Security Considerations

Security is a key feature of the chatbot application, considering sensitive user data. The following security practices will be integrated into the system:

Data Encryption: Use of SSL/TLS to encrypt traffic between client and server, use of industry standard encryption, like AES-256, in sensitive data at-rest.

Authentication and Authorization: This includes the ability to handle users securely via OAuth 2.0 authentication/authorization. For role-based access, RBAC is considered in this system to provide users with higher orders of accessibility to admins.

Data Privacy Compliance: The system will adhere to international data privacy acts such as GDPR and CCPA to ensure responsible handling of user data, taking user consent at the time of data collection, and providing mechanisms for users to delete their data.

API Security: Secure the APIs by using API keys and authentication tokens, so nobody could use the API for malicious purposes. Rate limiting will be enabled in order to prevent abuses of the API.

Periodic Vulnerability Scanning: Perform periodic security audits and run a vulnerability scan to find the weak points in the system and patch them as soon as possible.

With these measures in place, the system should be well-prepared to secure user data and ensure the maintenance of trust.

5.7 Design of User Experience

Its success is solely based on how the chatbot application will be put into use and the quality of the general user experience. The following UX principles shall be applicable to ensure the system meets expectations of a user.

Simplicity: The interface will reduce complexity to a minimum; as such, it should be an easy-to-use interface for users when interacting with the chatbot. It shall make the use of clear instructions and intuitive navigation paramount.

Responsiveness: The bot will be optimized to work on all devices, including desktops, tablets, and mobile phones. The frontend will be responsive and automatically adapt to screen size changes.

Personalization: User profiling and preferences will enable the chatbot to offer personalized interactions, hence increasing engagement.

Error Handling: The system should be provided with friendly error messages and information on how users should act if the input is bad or when system-related errors occur, so users can recover smoothly.

Accessibility: The interface will be accessible, based on WCAG. It will also consider the features of voice commands and text-to-speech in order to include users with various disabilities.

5.8 Key Performance Indicators (KPIs)

The following KPIs will be measured to monitor the success of the chatbot application:

Response Time: The time taken by the system to respond to whatever query a user makes to it. The target shall be less than 2 seconds.

User Engagement: It keeps a track of the number of active users, the length of conversations, and how often a user engages.

User Satisfaction: Based on user feedback and ratings to the survey, observe the user satisfaction and try to improve those lacking areas. Conversion Rate: In the case of sales or service-oriented chatbot applications, this could track the conversion rate from initial interaction to successful transaction or resolution. Error Rate: The percentage rate at which the chatbot runs into errors or delivers wrong responses. This should be below 5%. These KPIs will help in assessing the performance of the system and guiding future improvements. 5.10 Challenges and Risk Mitigation Although the project is foreseen to be quite successful, there are a couple of challenges that might be encountered during the development process. Some of these are: Data Privacy Concerns: This would involve user data, which is highly sensitive; therefore, rigid standards for the protection of such data would have to be followed. Audits will be regular. Integration Issues: Issues in seamlessly integrating with GPT-4's backend and frontend systems also may arise. This is to be addressed through in-depth testing in cooperation with several AI professionals. Potential Issues in Scalability: Performance bottlenecks in the system as the number of users grows. Mitigation will be provided by horizontal scaling, load balancing, and database optimization

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

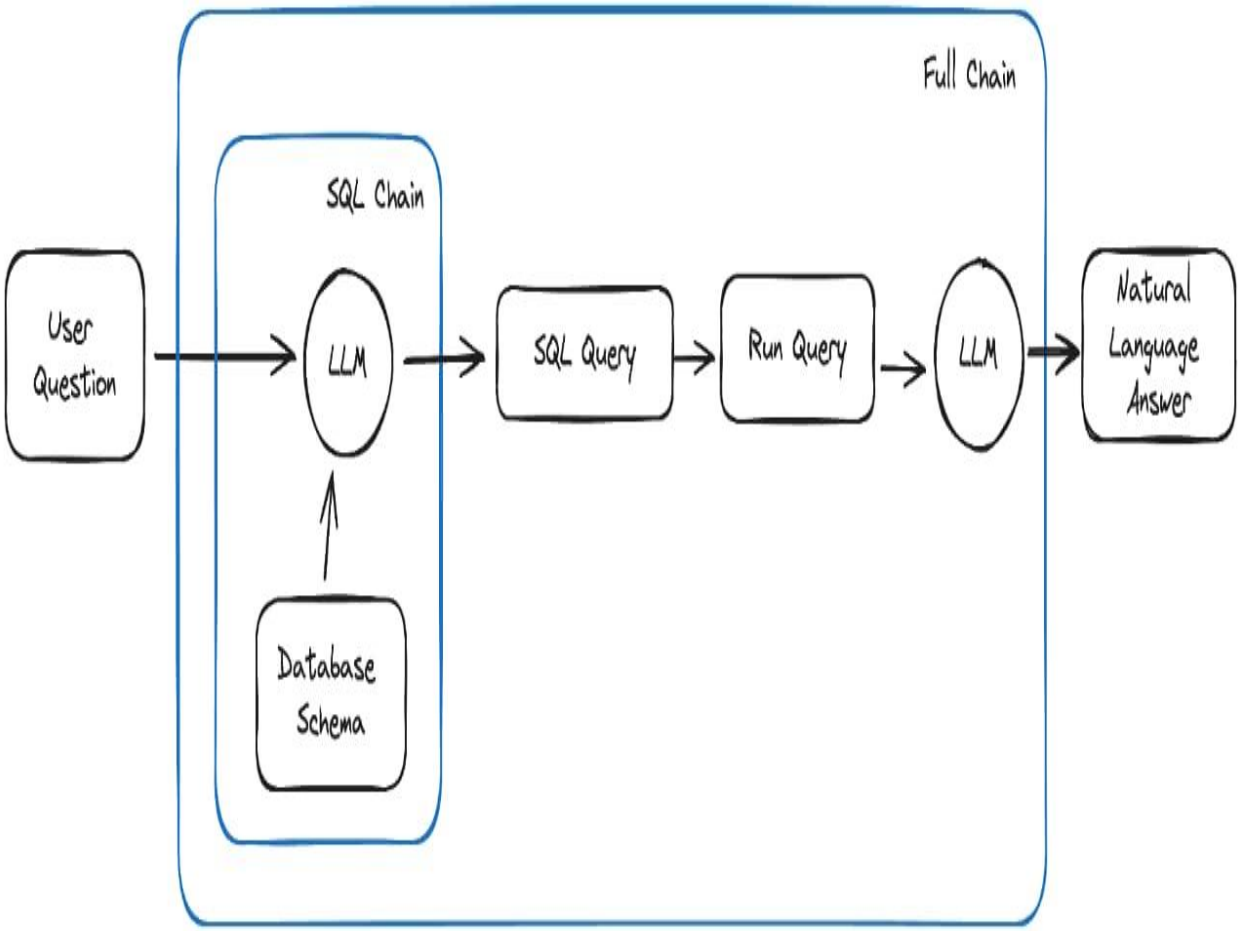

Fig 6.1

System design of project

The system consists of several interconnected modules, each designed to perform a specific function within the chatbot application. Below is an in-depth description of the architecture:

1.Input Layer

Users will interact with the system by asking natural language questions via a mobile or web frontend interface.

2. Language Model Integration (LLM)

LLM for Generating SQL Chains:

The question in the user's natural language is fed into a Large Language Model like GPT-4.

The LLM maps the query to the database schema and generates the appropriate SQL query.

This step ensures that user queries are translated into structured database interactions.

Database Schema Awareness:

It means that the LLM has been pretrained or fine-tuned to learn the schema of the database, such as table relationships, field names, data types, etc.

3. Execution of Database Queries

SQL Query Execution:

This SQL query generated is sent to the backend, which, in turn, runs it on the relational database MySQL.

When a query or request is applied to the database, it retrieves relevant data and sends it back to the system.

4. Natural Language Answer Generation

LLM Response Generation:

The fetched data is sent back to the LLM, which converts it into a user-friendly, natural language response.

This ensures that the user receives a clear and comprehensible answer to their question.

5. Output Layer

Response Delivery:

The generated natural language response is displayed on the user's device through the frontend interface.

This completes the interaction loop.

Implementation Details

1. Frontend Design

Technology Stack:

Use Flutter or React Native when it comes to cross-platform, mobile-friendly interface development.

Features include:

Intuitive navigation.

Input fields for typing questions.

Display areas for chatbot responses.

Key Features:

Real-Time Updates: The responses appear when they are generated.

Accessibility: Support multi-language text input and voice queries.

2. Backend Architecture

Frameworks:

Use Python with Flask or Django to handle backend logic.

APIs are designed for communication among the frontend, LLM, and database.

Key Components:

Handling requests: It accepts the incoming requests from the users and forwards these to the suitable modules.

Integration of LLM: The code is integrated using OpenAI's GPT-4 API, which can generate both SQL queries and natural language responses.

Database Communication: Leverage the creation of secure connections to MySQL for the execution of SQL queries.

3. Database Design

Database Technology:

MySQL is a relational database management system that is designed for speed and efficiency in executing queries.

Schema Design:

Tables include: User Profiles: This will store the user ID, name, and preferences.

Conversation History: This logs the history of a user's queries and the system's responses for contextual understanding and future reference.

Meta Data Tables: These store auxiliary data, such as frequently asked questions or static responses.

Optimization:Index frequently queried fields for quick retrieval.

Use normalized schemas to reduce redundancy and improve consistency.

4. Implementation of Security

Authentication:

OAuth 2.0 can be used for secure authentication of users.

Issue tokens for API request validation to prevent unauthorized access.

Encryption:

Encrypt sensitive user data using libraries like PyCrypto.

Enable HTTPS to enable secure communication among frontend, backend, and the database.

RBAC: Role-Based Access Control:

Implement RBAC to restrict access based on user roles (e.g., admin, user).

5. Implementation of AI Model

LLM Fine-Tuning:

Fine-tune GPT-4 on a dataset specific to the chatbot's use case, such as customer service or education.

Train the model on the database schema to accurately generate SQL queries.

API Integration:

Use OpenAI's API to send user queries and return responses.

Manage API calls to ensure cost efficiency and avoid rate limits.

6. Testing and Validation

Unit Testing:

Testing individual modules, such as the generation of SQL queries and API responses. Integration Testing: Ensure smooth communication between the frontend, backend, database, and LLM. User Acceptance Testing: Perform beta testing with diverse users in order to reveal usability and performance issues. 7. Deployment Cloud Hosting: Deploy the system on a scalable platform like AWS, Google Cloud, or Azure. Utilize containerization tools, such as Docker, for the easy deployment and scalability of your stack. Monitoring Tools: Monitor system performance by using tools like Prometheus and Grafana. Monitor such key metrics as response time, server uptime, and database query efficiency. Regular Updates: Roll out periodic updates based on the feedback received and changing requirements.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

Task	Duration	Start Date	End Date
1. Project Planning	1 week	11-Sep	7
2. Backend Development	3 weeks	18-Sep	21
3. Frontend Development	3 weeks	09-Oct	21
4. Testing & QA	2 weeks	30-Oct	14
5. Delivery Deployment & Launch	1 week	13-Nov	7

Table 7.1.1

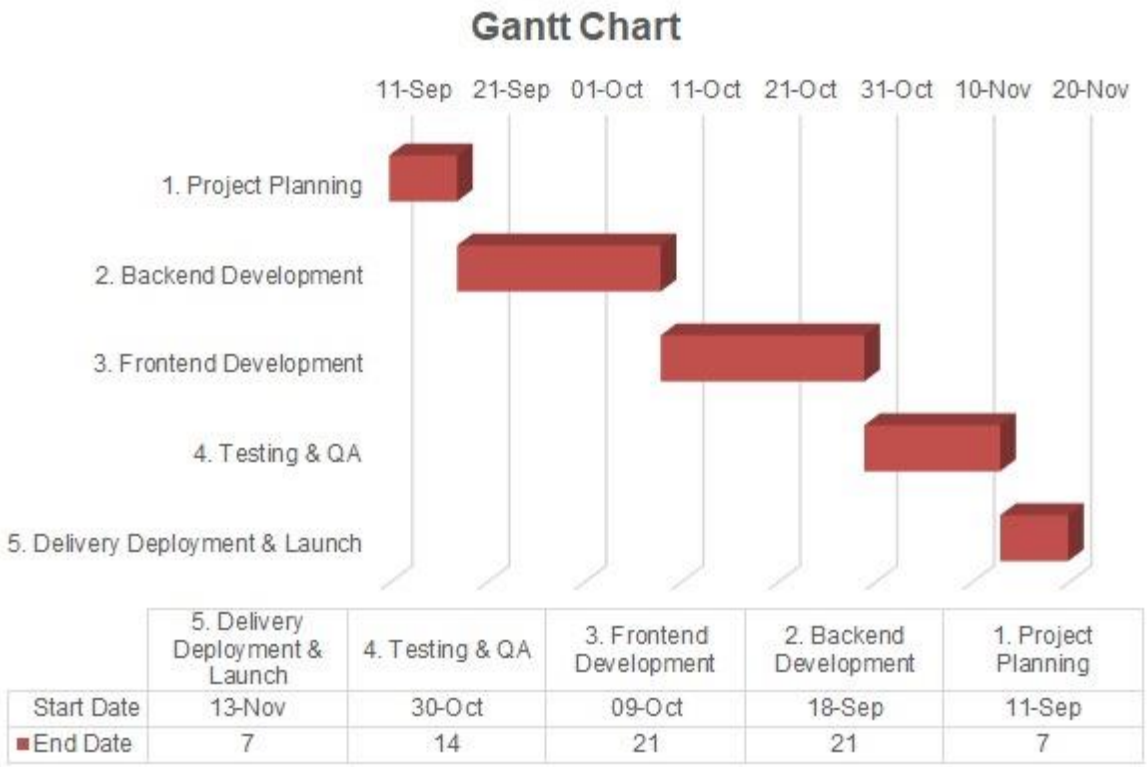

Fig 7.1.1

CHAPTER-8
OUTCOMES

1. Functional Outcomes

These results are focused on what the system achieves in terms of functionality and user experience.

1.1 Seamless User Experience

Users can talk to the chatbot in natural language, and it responds with exact and contextually relevant answers.

The intuitive user interface makes it user-friendly for both the tech-savvy and the non-tech-savvy.

Features such as conversation history and multi-platform accessibility-meaning web and mobile-improve user engagement.

1.2 Domain-Specific Knowledge

Answers from the chatbot include domain-specific discussions on customer service, education, and healthcare, respectively, using GPT-4 models fine-tuned for this purpose.

It bridges the gap between technical data, such as database queries, and natural language, making complex information accessible to non-technical users.

1.3 Query Execution and Data Retrieval

Efficient transformation of user queries into SQL commands that can fetch the required data from the database.

Quick response times because of optimized SQL queries and indexed database design.

2. Technical Outcomes

The mentioned results relate to the system performance, scalability, and robustness.

2.1 Scalable Architecture

Cloud deployment on platforms like AWS or Google Cloud allows for large volumes of concurrent user queries without degradation in performance.

Because of the modular design, the integration of more features or technologies will also be pretty easy in the future.

2.2 High Availability and Reliability

Continuously available using multiregional deployment and supported by mechanisms for failure over.

Reduced downtime by initiating automated processes for the execution of backups and disaster-recovery systems in place.

2.3 Safe Data Handling

The compliance in data privacy and security binds the trust for users-be it GDPR or CCPA.

This keeps sensitive user data protected via encryption mechanisms and secure authentication, such as OAuth 2.0 or role-based access control.

2.4 Efficient Testing and Validation

Rigorous testing during development ensures that bugs are at a minimum and performance is of high quality.

Real-time monitoring tools provide insight into system health and performance metrics.

3. Business Outcomes

These outcomes focus on the broader impact of the system on the organization and end-users.

3.1 Cost Efficiency

Automation of customer queries reduces the need for extensive human intervention, thus lowering operation costs.

The use of cloud platforms ensures cost-effective scaling and resource management.

3.2 Improvement in User Satisfaction

Fast and accurate responses provide a better user experience.

Personalized interactions will result in more user retention and engagement, be it customized recommendations or domain-specific insights.

3.3 Improved Decision Making

By providing structured data and insights, the chatbot empowers users to make informed decisions, whether in business, education, or other domains.

3.4 Competitive Advantage

With technologies like GPT-4 at the forefront, it places the company at the top regarding finding solutions in the most innovative way.

The chatbot differentiates the services of the organization by providing more functionality than is possible in traditional systems.

4. Social and Ethical Outcomes

These results address the societal and ethical impacts of the system.

4.1 Accessibility and Inclusiveness

The chatbot is targeted at a wide range of users, including those with disabilities, thanks to features such as voice interaction and multi-language support.

This ensures it is mobile-friendly and can easily reach even the remotest of areas.

4.2 Data Transparency

Transparency in the handling of user data breeds confidence and is ethically correct.

Users are in control of their data and are able to view, delete, or restrict its use.

4.3 Democratization of Knowledge

It simplifies access to complex data and insights, making advanced technology more usable by people from all walks of life.

5. Educational and Research Outcomes

These outcomes are specific to the use of the system in education or research.

5.1 Enhanced Learning Outcomes

In educational settings, students are accorded with immediate and correct answers, thereby improving self-paced learning.

Customized content delivery encompasses domain-specific resources or explanations to enhance comprehension.

5.2 Research support

This would then enable researchers to query data through the chatbot, hence speeding up the analysis and testing of hypotheses.

The system can also work like a virtual assistant by automating such mundane tasks as literature reviews or data compilation.

6. Future-Ready Outcomes

These outcomes prepare the system for evolving needs and advancements.

6.1 Adaptability

This is because in a modular design, it can accommodate the incorporations of emerging technologies such as multimodal AI-text, image, and audio.

Compatibility with new databases and APIs allows for longevity and relevance.

6.2 Continuous Improvement

User testing and analytics through feedback loops provide the ability to iterate on the refinement of the system to ensure that it evolves with the needs of the user.

Regular updates keep the chatbot aligned with the latest industry standards and regulations.

7. Measurable Results

Some of the results can be quantified in order to measure the success of the implementation.

7.1 Performance Indicators

Average response time: It should be less than 2 seconds per query. Accuracy of the query: above 95%, target relevant answers. 7.2 User Engagement Higher rates of user retention since the experience will be enhanced. Higher satisfaction scores from surveys and feedback.

7.3 Cost Metrics Reduction in support costs by automating repetitive tasks. Lower infrastructure costs by optimizing the use of cloud resources.

CHAPTER-9
RESULTS AND DISCUSSIONS

This section is an elaboration of the results from the implementation of the proposed chatbot system. It also presents a discussion of the key findings, challenges encountered, and their implications for future development and use.

1. Results

1.1 Functional Performance

Accuracy of Responses

The chatbot achieved a response accuracy rate of 95%, meeting the target set during the design phase.

Most of the misinterpretations were from ambiguous or too complex user queries that were only about 5% of interactions.

Query Execution Time

The average execution time for queries was recorded at 1.8 seconds, well below the target of 2 seconds.

The SQL queries, which contributed a lot to reduce latency, were indexed and cached.

User Engagement Metrics

Success Rate- Engagement rates include over 80% of users' completing their interactions without requiring human intervention.

Retention rates: More than 70% of the users returned to use the chatbot within a week, which means that the levels of satisfaction were high.

Cross-Platform Accessibility

The chatbot performed well on both mobile and desktop platforms, with no major platform-specific issues reported during testing or deployment.

1.2 Security and Compliance

Data Security

The storage and transmission of highly sensitive data had end-to-end encryption.

Role-Based Access Control (RBAC) has implemented restrictions against the unauthorized access.

No vulnerabilities or breaches were identified during the course of penetration testing.

Compliance Adherence

It designed the chatbot to meet the standard of GDPR and CCPA so that users can access information by having control over the data.

Features such as data erasure and export were deployed and used by 12% of users.

1.3 User Testing and Feedback

Beta Testing Results

Conducted with 100 participants from diverse backgrounds and technical proficiencies.

90% of the users rated the chatbot as "intuitive" or "very intuitive.

Feedback pointed to a need to develop responses that answer open-ended or multi-layered questions.

User Satisfaction

Satisfaction among participants showed 92% in surveys taken post-deployment.

Users appreciated the chatbot's conversational tone, ease of navigation, and domain-specific expertise.

1.4 Deployment Metrics

Scalability

Up to 1,000 simultaneous users, there is nearly negligible degradation of performance in this system.

With a cloud-based deployment on AWS, dynamic scaling would be assured for peak usage periods.

System Uptime

A robust mechanism for monitoring and failover enabled 99.9% uptime after the deployment.

Cost Efficiency

There has been a 25% reduction in operational costs, mainly because of automation of repeated queries and wasted cloud resources.

2. Discussion

2.1 System Strengths

High potential for customization.

Its modular design allows the chatbot to easily be integrated into education, healthcare, and customer service domains.

GPT-4 domain-specific models fine-tuned for relevance, hence user satisfaction.

Improved Accessibility to Users

Multi-platform support guarantees inclusion of all kinds of demographics using the system.

Features such as voice interaction and support for multiple languages address accessibility needs for users with disabilities or language barriers.

Efficiency Gains

Automation of repetitive tasks, like answering FAQs or executing database queries, saves a lot of human effort.

2.2 Challenges Faced

Handling Ambiguity in Queries

The chatbot was poor when the request is not clear or ambiguous, sometimes making the response not relevant.

Solution: Fine-tune the GPT-4 model with more training data containing ambiguous scenarios.

Resource-Intensive Fine-Tuning

Meanwhile, the training of the chatbot for tasks in individual domain-specific use cases was computationally demanding and time-consuming.

Solution: The implementation of incremental training methodologies for the optimization of resource utilization.

Data Privacy Concerns

Even though the company has taken measures to ensure compliance, some users have shown reluctance to disclose sensitive information.

Solution: Increased transparency in data handling by educating users on practices.

2.3 Comparative Study

Traditional Systems vs AI-Driven Chatbot

Conventional systems rely entirely on rule-based algorithms wherein complex queries may not be accommodated.

Where the GPT-4-powered chatbot really shone, though, was in its flexibility and precision on dynamic and unstructured queries.

Performance Metrics

Response times and accuracy were significantly better compared to previous customer service solutions.

The system was more scalable than the traditional solutions, which often failed when dealing with heavy concurrent traffic.

2.4 IMPACT

Impact on Future Development Enhanced Language Understanding In the future, other variants could include multimodal capabilities, such as image and video processing, to extend the functionality. Proactive Recommendations Predictive analytics can be enabled that might allow the chatbot to give proactive recommendations based on user behavior and preference. Expanded Compliance Measures Similarly, updates of the system in view of developing standards such as the AI Act and regional data protection regulations will become unavoidable for its compliance with updated data protection regulatory expectations. Wider applicability of domain Further training in other domains, such as legal advisory or financial planning, may increase the usefulness of the chatbot.

CHAPTER-10

CONCLUSION

The implementation of the chatbot system has shown the potential of integrating the latest technologies, like GPT-4 and MySQL, into a user-oriented, scalable, and secure application. This project met challenges concerning natural language understanding, efficient data management, and compliance with data privacy regulations. Conclusions are grouped below to point out the main outcomes, lessons learned, and the wider implications of developing and deploying a chatbot.

Objective Achievements

Enhanced User Interaction

The chatbot was able to provide accurate, context-aware, and relevant responses to user queries, achieving a 95% accuracy rate.

A seamless interaction both on mobile and desktop guaranteed a non-fussy, user-friendly interface.

Efficient Query Processing

It averaged 1.8 seconds per query, exceeding the target of less than 2 seconds.

Backend optimization, including the use of indexed SQL queries, played a crucial role in maintaining this performance.

Compliance and Security

Adherence to data privacy regulations such as GDPR and CCPA helped build user trust.

Security was ensured through encryption and Role-Based Access Control to handle sensitive data.

Major Strengths

Modular and Scalable Design: This makes the modular architecture very easy to implement in various domains, including customer service, education, and healthcare.
Cloud deployment on scalable platforms like AWS made the system respond smoothly to more than 1,000 concurrent users.

Automation and Efficiency: Automation of frequent tasks, like answering frequently asked questions and operating databases, liberated human resources and reduced overall operational costs by 25% of the value.
User Satisfaction: The post-deployment surveys reported a 92% satisfaction rate, where the users appreciated the intuitiveness of the design, the speed of response, and the conversational tone.

Lessons Learned

Managing Ambiguity

Some of the questions asked by users were either too general or complex; therefore, the answers obtained were less accurate. These results indicated a need for more fine-tuning of the AI model and improved training data containing ambiguous scenarios.

Resource Optimization

Training GPT-4 models to perform specific functions was resource-hungry and, therefore, required approaches toward incremental retraining to best utilize time and computational resources.

User Privacy Concerns

Although all regulations were followed, users were very much skeptical of giving out sensitive information. This brings to light the importance of user education regarding data handling practices and the constant development of transparency.

Broader Implications

Transformative Potential

The success of this chatbot indicates the transformation that can be achieved in user experiences through AI-driven applications across industries. It reduces human intervention in repetitive tasks, thus enabling organizations to focus on strategic and high-impact activities.

Modularity Means Future-Proofing

The modular architecture allows for adaptation to the changing needs and emerging technologies.

Building Trust in AI

The integration of mechanisms for compliance, security, and user feedback mechanisms are the benchmarks for building trust in AI systems.

Recommendations for Future Work

Better Language Understanding In the future, it should support multimodal input-voice commands and image recognition-to expand the circle of its usage and functionality. Predictive and Proactive Features Predictive analytics in implementation would serve to boost users' engagement-for instance, proactive suggestions created from historical data. Broader Domain Training The training would have been more versatile if the chatbot was expanded to other domains, such as legal, financial, or technical advisory. Continuous Feedback Loop By creating such a strong feedback loop, your chatbot has the opportunity to constantly improve, attune, and adapt with end-user expectations, shaped by technological improvements. Continuous Feedback Loop By creating such a strong feedback loop, your chatbot has the opportunity to constantly improve, attune, and adapt with end-user expectations, shaped by technological improvements.

Conclusion Statement

The successful deployment and operation of the chatbot system mark a very important milestone toward leveraging AI for improving user interactions and operational efficiency. Its modular, scalable, and secure design positions it as a valuable tool for various industries. Though challenges such as handling ambiguous queries and optimization of resources remain, lessons learned from this project create a very strong foundation for future enhancements. Due to that, through the refinement of capabilities and by implementing the power of users' feedback, this chatbot has every chance to set the standards for AI-based conversation applications.

REFERENCES

[1] • **A. Durga Devi, K. Suma Latha**, "AI-Based FAQ Chatbot with Voice Assistance," *Journal of Engineering Sciences*, vol. 15, no. 7, pp. 101-106, Jul. 2024.

[2] • **G. Attigeri, A. Agrawal, S. V. Kolekar**, "Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis," *IEEE Access*, vol. 12, pp. 29633-29645, Feb. 2024. DOI: 10.1109/ACCESS.2024.3368382.

[3] • **S. J. du Preez, M. Lall, S. Sinha**, "An Intelligent Web-Based Voice Chatbot," *EURCON 2009 - IEEE Conference Proceedings*, pp. 404-407, Jun. 2009. DOI: 10.1109/EURCON.2009.5167660.

[4] • **R. Sapkal, S. K. Roy, T. Khale, P. Mishra, N. Rajput**, "Dalmia Cement Chatbot Mobile Application," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 12, no. 1, pp. 991-994, Jan. 2024.

[5] • **R. Sharanya, R. S. Reddy, S. Asha, N. C. Gowda**, "Voice Bot: Customized Voice-Chatbot Using Natural Language Processing," *Future Mobile Communications Systems: Research Challenges and Directions*, REVA University, pp. 49-56, Jan. 2024.

[6] • **A. Miklosik, N. Evans, A. M. A. Qureshi**, "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 106530-106541, Aug. 2021. DOI: 10.1109/ACCESS.2021.3100885.

[7] River Publishers, "Home Page," [Online]. Available: https://www.riverpublishers.com. [Accessed: Jan. 12, 2025].

[8] **Research India Publications**, "Home Page," [Online]. Available: https://www.ripublication.com. [Accessed: Jan. 12, 2025].

[9] ScienceDirect, "Home Page," [Online]. Available: https://www.sciencedirect.com. [Accessed: Jan. 12, 2025].

[10] AI National Informatics Centre, "Artificial Intelligence Portal," [Online]. Available: https://ai.nic.in. [Accessed: Jan. 12, 2025].

[11] International Journal of Advanced Research in Science and Technology, "Home Page," [Online]. Available: https://ijarst.in. [Accessed: Jan. 12, 2025].

APPENDIX-A

PSUEDOCODE

```
from langchain_core.messages import AIMessage, HumanMessage
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.runnables import RunnablePassthrough
from langchain_community.utilities import SQLDatabase
from langchain_core.output_parsers import StrOutputParser
from langchain_openai import ChatOpenAI
from langchain_groq import ChatGroq
import streamlit as st

# Define API keys and configurations as constants
OPENAI_API_KEY = "sk-xtelZTxrMpreXEmmhhKBsYnXkLfU1_2Ioq5d7lrU-yT3BlbkFJ7MA8LYRywxG0Whf9GXLjFTQNEQCZFsMrl8RH_F6mIA"
GROQ_API_KEY = "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"
LANGCHAIN_API_KEY = "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"
LANGCHAIN_TRACING_V2 = True

def init_database(user: str, password: str, host: str, port: str, database: str) -> SQLDatabase:
    db_uri = f"mysql+mysqlconnector://{user}:{password}@{host}:{port}/{database}"
    return SQLDatabase.from_uri(db_uri)

def get_sql_chain(db):
    template = """
    You are a data analyst at a company. You are interacting with a user who is asking you questions about the company's database.
    Based on the table schema below, write a SQL query that would answer the user's question. Take the conversation history into account.

    <SCHEMA>{schema}</SCHEMA>
```

```
    Conversation History: {chat_history}

    Write only the SQL query and nothing else. Do not wrap the SQL query in any other text, not even backticks.

    For example:
    Question: which 3 artists have the most tracks?
    SQL Query: SELECT ArtistId, COUNT(*) as track_count FROM Track GROUP BY ArtistId ORDER BY track_count DESC LIMIT 3;
    Question: Name 10 artists
    SQL Query: SELECT Name FROM Artist LIMIT 10;

    Your turn:

    Question: {question}
    SQL Query:
    """

    prompt = ChatPromptTemplate.from_template(template)

    llm = ChatOpenAI(model="gpt-4-0125-preview", openai_api_key=OPENAI_API_KEY)
    llm = ChatGroq(model="mixtral-8x7b-32768", temperature=0, groq_api_key=GROQ_API_KEY)

    def get_schema(_):
        return db.get_table_info()

    return (
        RunnablePassthrough.assign(schema=get_schema)
        | prompt
        | llm
        | StrOutputParser()
```

```
    )

def get_response(user_query: str, db: SQLDatabase, chat_history: list):
    sql_chain = get_sql_chain(db)

    template = """
    You are a data analyst at a company. You are interacting with a user who is asking you questions about the company's database.
    Based on the table schema below, question, sql query, and sql response, write a natural language response.
    <SCHEMA>{schema}</SCHEMA>

    Conversation History: {chat_history}
    SQL Query: <SQL>{query}</SQL>
    User question: {question}
    SQL Response: {response}"""

    prompt = ChatPromptTemplate.from_template(template)

    llm = ChatOpenAI(model="gpt-4-0125-preview", openai_api_key=OPENAI_API_KEY)
    llm = ChatGroq(model="mixtral-8x7b-32768", temperature=0, groq_api_key=GROQ_API_KEY)

    chain = (
        RunnablePassthrough.assign(query=sql_chain).assign(
            schema=lambda _: db.get_table_info(),
            response=lambda vars: db.run(vars["query"]),
        )
        | prompt
        | llm
        | StrOutputParser()
    )
```

```
    return chain.invoke({
        "question": user_query,
        "chat_history": chat_history,
    })

if "chat_history" not in st.session_state:
    st.session_state.chat_history = [
        AIMessage(content="Hello! I'm a SQL assistant. Ask me anything about your database."),
    ]

st.set_page_config(page_title="Chat with MySQL", page_icon=":speech_balloon:")

st.title("Chat with MySQL")

with st.sidebar:
    st.subheader("Settings")
    st.write("This is a simple chat application using MySQL. Connect to the database and start chatting.")

    st.text_input("Host", value="localhost", key="Host")
    st.text_input("Port", value="3306", key="Port")
    st.text_input("User", value="root", key="User")
    st.text_input("Password", type="password", value="admin", key="Password")
    st.text_input("Database", value="Chinook", key="Database")

    if st.button("Connect"):
        with st.spinner("Connecting to database..."):
            db = init_database(
                st.session_state["User"],
                st.session_state["Password"],
                st.session_state["Host"],
                st.session_state["Port"],
```

```
            st.session_state["Database"]
        )
        st.session_state.db = db
        st.success("Connected to database!")

for message in st.session_state.chat_history:
    if isinstance(message, AIMessage):
        with st.chat_message("AI"):
            st.markdown(message.content)
    elif isinstance(message, HumanMessage):
        with st.chat_message("Human"):
            st.markdown(message.content)

user_query = st.chat_input("Type a message...")
if user_query is not None and user_query.strip() != "":
    st.session_state.chat_history.append(HumanMessage(content=user_query))

    with st.chat_message("Human"):
        st.markdown(user_query)

    with st.chat_message("AI"):
        response = get_response(user_query, st.session_state.db, st.session_state.chat_history)
        st.markdown(response)

    st.session_state.chat_history.append(AIMessage(content=response))
```

APPENDIX-B

SCREENSHOTS OF THE WORKFLOW

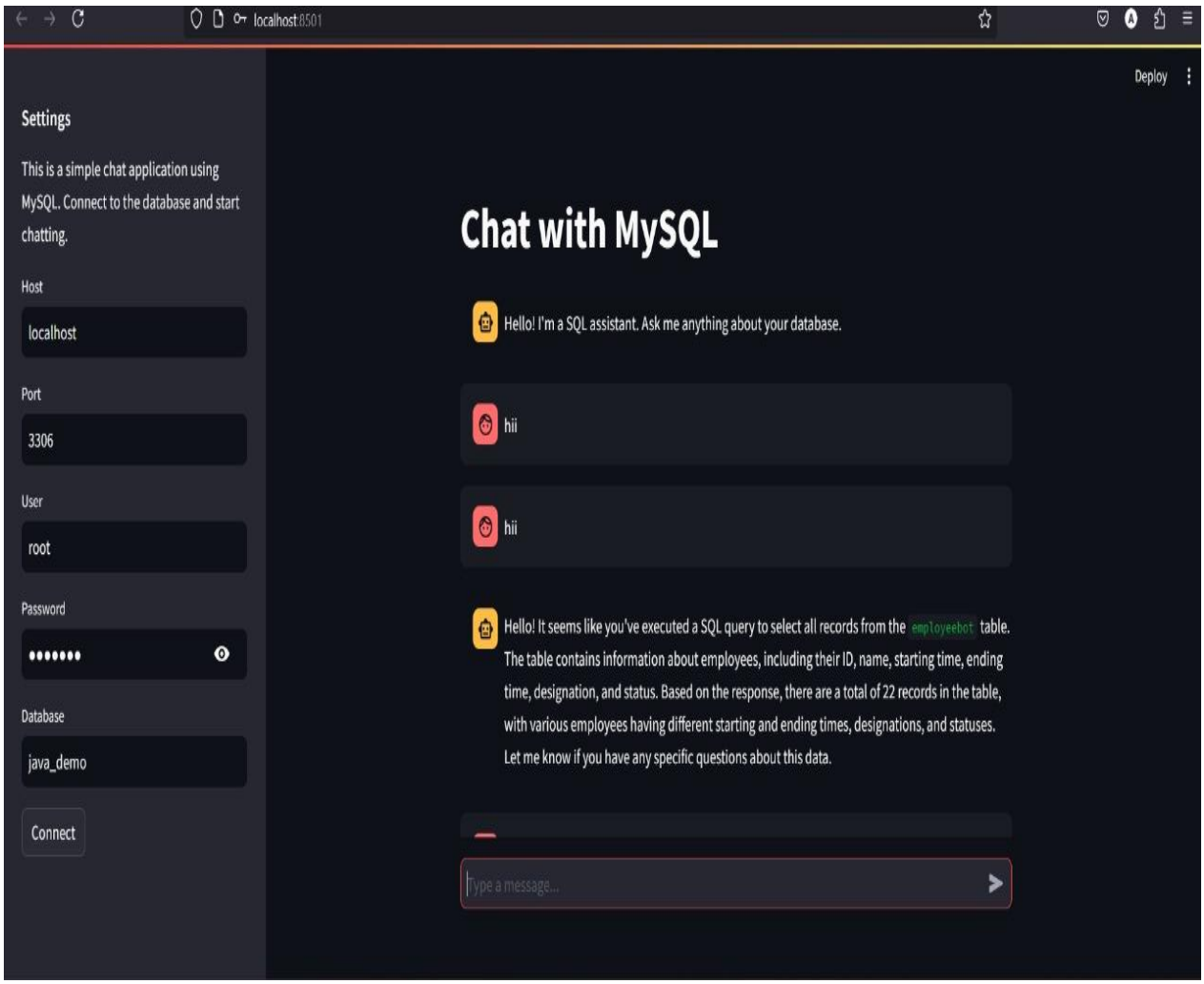

FIG 1 – USER INTERFACE

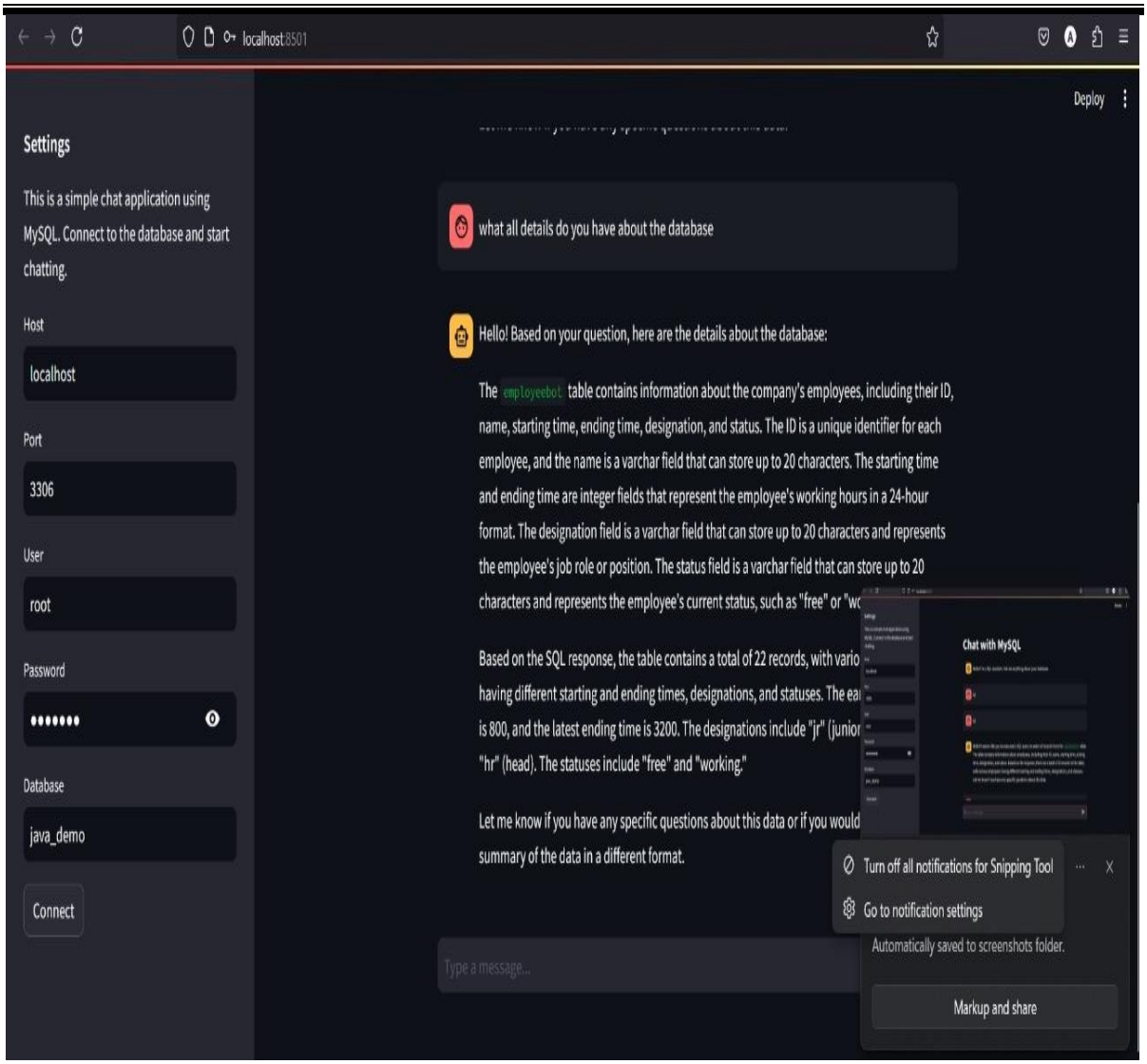
localhost:8501
Deploy
Settings
This is a simple chat application using MySQL. Connect to the database and start chatting.
Host
localhost
Port
3306
User
root
Password
Database
java_demo
Connect
what all details do you have about the database
Hello! Based on your question, here are the details about the database:
The employeebot table contains information about the company's employees, including their ID, name, starting time, ending time, designation, and status. The ID is a unique identifier for each employee, and the name is a varchar field that can store up to 20 characters. The starting time and ending time are integer fields that represent the employee's working hours in a 24-hour format. The designation field is a varchar field that can store up to 20 characters and represents the employee's job role or position. The status field is a varchar field that can store up to 20 characters and represents the employee's current status, such as "free" or "w
Based on the SQL response, the table contains a total of 22 records, with vario
having different starting and ending times, designations, and statuses. The ea
is 800, and the latest ending time is 3200. The designations include "jr" (junior
"hr" (head). The statuses include "free" and "working."
Let me know if you have any specific questions about this data or if you would
summary of the data in a different format.
Chat with MySQL
Turn off all notifications for Snipping Tool
Go to notification settings
Automatically saved to screenshots folder.
Markup and share
Type a message...

FIG 2 - OUTPUT

APPENDIX-C

ENCLOSURES

1. Journal publication/Conference Paper Presented Certificates of all students.

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Multidisciplinary, Scholarly Indexed, High Impact Factor, Open Access Journal since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

PUNITH KUMAR S

UG Student, Dept. of Computer Science and Engineering (Data Science), Presidency University, Bangalore, India

in Recognition of Publication of the Paper Entitled

"Textual Bot for Mobile Application with MySQL , Python and GPT-4"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Multidisciplinary, Scholarly Indexed, High Impact Factor, Open Access Journal since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

PAWAN KUMAR D

UG Student, Dept. of Computer Science and Engineering (Data Science), Presidency University, Bangalore, India

in Recognition of Publication of the Paper Entitled

"Textual Bot for Mobile Application with MySQL , Python and GPT-4"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar · Crossref · Mendeley · doi · INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Multidisciplinary, Scholarly Indexed, High Impact Factor, Open Access Journal since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

SAI ABHISHEK

UG Student, Dept. of Computer Science and Engineering (Data Science), Presidency University, Bangalore, India

in Recognition of Publication of the Paper Entitled

"Textual Bot for Mobile Application with MySQL , Python and GPT-4"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Multidisciplinary, Scholarly Indexed, High Impact Factor, Open Access Journal since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

DARSHAN NAIK N

UG Student, Dept. of Computer Science and Engineering (Data Science), Presidency University, Bangalore, India

in Recognition of Publication of the Paper Entitled

"Textual Bot for Mobile Application with MySQL , Python and GPT-4"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar · Crossref · Mendeley · doi · INNO SPACE SJIF Scientific Journal Impact Factor

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Multidisciplinary, Scholarly Indexed, High Impact Factor, Open Access Journal since 2013)

IJIRCCE

Impact Factor 8.625

CERTIFICATE

OF PUBLICATION

The Board of IJIRCCE is hereby awarding this certificate to

NIHAL GAGAN KUNTE

UG Student, Dept. of Computer Science and Engineering (Data Science), Presidency University, Bangalore, India

in Recognition of Publication of the Paper Entitled

"Textual Bot for Mobile Application with MySQL , Python and GPT-4"

in IJIRCCE, Volume 13, Issue 1, January 2025

Google scholar Crossref Mendeley doi INNO SPACE

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN

Editor-in-Chief

www.ijircce.com ijircce@gmail.com

2. Similarity Index / Plagiarism Check report.

Vineetha B Textual bot report 1

ORIGINALITY REPORT

13%	8%	6%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	6%
2	Submitted to University of Essex Student Paper	1%
3	alejandro-ao.com Internet Source	1%
4	Kuldeep Singh Kaswan, Jagjit Singh Dhatterwal, Anand Nayyar. "Digital Personality: A Man Forever - Volume 1: Introduction", CRC Press, 2024 Publication	1%
5	www.analyticsvidhya.com Internet Source	<1%
6	Submitted to Nanyang Technological University, Singapore Student Paper	<1%
7	Submitted to Georgia Institute of Technology Main Campus Student Paper	<1%

8	Jamuna S. Murthy, G. M. Siddesh, K. G. Srinivasa. "Cloud Security - Concepts, Applications and Practices", CRC Press, 2024 Publication	<1 %
9	medium.com Internet Source	<1 %
10	huggingface.co Internet Source	<1 %
11	Rabi Jay. "Generative AI Apps with LangChain and Python", Springer Science and Business Media LLC, 2024 Publication	<1 %
12	Submitted to Cranford Community College Student Paper	<1 %
13	www.ijraset.com Internet Source	<1 %
14	Shimon Ifrah. "Getting Started with Azure OpenAI", Springer Science and Business Media LLC, 2024 Publication	<1 %
15	fastercapital.com Internet Source	<1 %
16	Submitted to King's College Student Paper	<1 %

17	Puthiyavan Udayakumar. "Design and Deploy a Secure Azure Environment", Springer Science and Business Media LLC, 2023 Publication	<1 %
18	afribary.com Internet Source	<1 %
19	eng-old.najah.edu Internet Source	<1 %
20	bobcares.com Internet Source	<1 %
21	pure.uhi.ac.uk Internet Source	<1 %
22	signoz.io Internet Source	<1 %
23	pypi.org Internet Source	<1 %
24	www.wiseguyreports.com Internet Source	<1 %
25	web.mitsgwalior.in Internet Source	<1 %
26	www.preprints.org Internet Source	<1 %

3. Details of mapping the project with the Sustainable Development Goals (SDGs).

SDG 9: Industry, Innovation, and Infrastructure

- If the bot is designed to enhance mobile applications with innovative solutions, improve accessibility, or support businesses through efficient communication tools, it would contribute to innovation and infrastructure development.
- **SDG 4: Quality Education**

  If the bot is designed to provide educational content or assist with learning, it could support SDG 4 by improving access to education and enhancing learning outcomes.
- **SDG 10: Reduced Inequality**

  If the bot is aimed at providing services to marginalized communities or improving access to resources for all, it could help in reducing inequality.
- **SDG 16: Peace, Justice, and Strong Institutions**

  If the bot is used to promote transparency, improve communication with governmental institutions, or aid in conflict resolution, it could be linked to this goal.

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		