International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

CHATBOTS IN EDUCATION SYSTEM

Gudivada Chitra Priyanka, D.Likhitha, A.Kamal Teja, J.Sirisha

ABSTRACT

This project presents the development of an advanced chatbot tailored specifically for college students, leveraging cutting-edge natural language processing (NLP) technologies similar to those employed by ChatGPT.

The primary objective of this chatbot is to enhance student support and engagement by providing personalized assistance and information. The chatbot is designed to address a range of student needs, including academic support, and simplified LMS resources. Through integration with institutional databases, the chatbot delivers real-time answers to common queries, offers study tips.

The system utilizes machine learning algorithms to continuously improve its responses based on user interactions and feedback. This project aims to improve student satisfaction by providing an accessible, intelligent, and responsive support tool, ultimately fostering a more efficient and supportive educational environment.

By integrating this chatbot into the college's existing systems, the project seeks to streamline student services, reduce administrative burdens, and enhance the overall educational experience. The ultimate goal is to create a responsive, user-friendly tool that fosters a supportive and efficient learning environment.

Keywords- Artificial intelligence, Chatbot, Anthropomorphism, Social presence, Compliance

Customer service

CHAPTER -1

INTRODUCTION

1.1 INTRODUCTION

This proposed system aims to revolutionize the learning experience for college students by providing a personalized, interactive, and engaging platform tailored to individual needs. By leveraging advanced artificial intelligence and natural language processing techniques, the system will offer personalized learning paths, real-time feedback, and a wealth of educational resources.

- In the dynamic landscape of higher education, the need for efficient, accessible, and personalized student support systems has never been greater. As colleges and universities strive to enhance the student experience, the

integration of artificial intelligence (AI) and advanced technologies offers promising solutions. This project focuses on developing a sophisticated chatbot, inspired by models like ChatGPT, specifically designed to meet the diverse needs of college students.

- The primary goal of this chatbot is to provide a comprehensive support tool that enhances student engagement, streamlines administrative processes, and offers academic assistance. Traditional methods of student support—such as in-person consultations and email communications—can be time-consuming and often fall short of meeting the fastpacked demands of today's students. A well-designed chatbot can bridge this gap by offering instant, reliable, and context-aware responses to a wide range of student inquiries.

- By leveraging natural language processing and machine learning, the chatbot can deliver personalized responses and adapt to individual student needs over time.

- The implementation of this chatbot seeks to address common challenges faced by students and educational institutions alike. By providing a seamless, 24/7 support system, it aims to improve student satisfaction, reduce the administrative workload on staff, and create a more engaging and responsive educational environment. This project represents a significant step towards integrating AI-driven tools into the academic sphere, enhancing both the efficiency and effectiveness of student support services.

This system aims to revolutionize college education by providing personalized learning paths, real-time feedback, and curriculum alignment. It leverages AI to adapt to individual student needs, enhancing understanding and improving academic outcomes. The system integrates seamlessly with existing college systems, offering a convenient and effective learning experience.

This system aims to revolutionize college education by offering personalized learning paths, real-time feedback, and curriculum alignment. By leveraging AI and NLP, the system tailors content to individual student needs, enhancing understanding and improving academic outcomes. The system seamlessly integrates with existing college systems, providing a convenient and effective learning experience.

1.2 EXISTING SYSTEM

1.Natural Language Processing (NLP) Platforms: These are foundational for AI chatbots, enabling them to understand and generate human-like text. Systems like Google's Dialogflow, Microsoft's LUIS, or open-source options like Rasa are widely used. Compliance here involves ensuring that the NLP models are trained on appropriate data to avoid biases and meet regulatory standards.

2. **Cloud-Based Platforms:** Many chatbots leverage cloud services like AWS, Azure, or Google Cloud for scalability and integration capabilities. Compliance issues often revolve around data privacy (e.g., GDPR, CCPA) and data security (ensuring encryption and secure transmission of data).

3. **Integration with CRM Systems:** Chatbots often need to integrate with Customer Relationship Management (CRM) systems like Salesforce or HubSpot. Compliance involves handling customer data securely and ensuring data accuracy in automated interactions.

4.Machine Learning Models: Advanced chatbots may employ machine learning models for intent recognition, sentiment analysis, or personalized responses. Compliance challenges include transparency in how data is used for training and ensuring that decisions are explainable and fair.

5. Regulatory Compliance: Depending on the industry and geographical location, chatbots must adhere to various regulations such as GDPR in Europe, HIPAA in healthcare, or PCIDSS in payment processing. Compliance here involves data handling practices, user consent, and security measures.

Ethical Considerations: There's growing awareness of ethical AI usage, including avoiding bias, respecting user privacy, and providing clear disclosures about the use of AI in customer interactions. Compliance efforts include auditing AI models for biases and ensuring ethical guidelines are followed customer service standards.

1.2.1 DEMERITS OF EXISTING SYSTEM

Lack of Human Touch: AI chatbots, especially those based on rule-based systems or less sophisticated AI, may struggle to provide the empathy and nuanced understanding that human agents can offer. This can lead to customer frustration, especially in sensitive or complex situations.

Limited Contextual Understanding: AI chatbots might struggle with understanding context or interpreting ambiguous or colloquial language, leading to miscommunications or incorrect responses.

Dependency on Data Quality: AI chatbots heavily rely on the quality and relevance of the data they are trained on. Poor data quality can lead to inaccurate responses or biased outcomes.

Initial Setup and Maintenance Costs: Developing and deploying AI chatbots can be costly, especially when integrating with existing systems like CRM or ERP. Maintenance costs include updates to keep the chatbot's knowledge and responses up- to-date.

Security and Privacy Concerns: Storing and processing customer data poses security risks. Chatbots must adhere to stringent data protection regulations (e.g., GDPR) to avoid breaches and maintain user trust.

Complexity in Handling Complex Issues: While AI chatbots excel in handling routine queries, they may struggle with more complex issues that require human judgment, creativity, or deep domain knowledge.

User Resistance and Adoption: Some users may prefer human interaction over interactions with chatbots, leading to resistance or lower adoption rates among certain demographics or in specific industries.

Integration Challenges: Integrating AI chatbots with existing systems and workflows can be challenging, requiring substantial effort to ensure seamless operation and data flow.

1.3 PROPOSED SYSTEM

This proposed system aims to be a more effective and personalized learning companion for students compared to ChatGPT. It will leverage advanced AI techniques, tailored data, and real-time feedback to provide a superior learning experience.

Individualized Content: The system will adapt to each student's learning style, pace, and understanding level to create customized learning paths.

Adaptive Assessments: Continuous assessments will help identify knowledge gaps and adjust the learning path accordingly.

Immediate Feedback: The system will provide instant feedback on student responses, explaining errors and offering suggestions for improvement.

Intelligent Tutoring: The system will act as a virtual tutor, guiding students through complex concepts and answering their questions.

Curriculum Alignment: The system will be closely aligned with the college's curriculum, ensuring that students are learning the necessary material.

Supplementary Resources: It will provide additional resources, such as practice problems, videos, and articles, to complement classroom learning.

Deep Understanding: The system will go beyond simple keyword matching to understand the context of student queries and provide more relevant and informative responses.

Reasoning Abilities: It will be able to reason through problems, apply logical thinking, and make connections between different concepts.

Improved Language Understanding: The system will use state-of-the-art NLP techniques to better comprehend and respond to student questions in natural language.

Contextual Awareness: It will be able to maintain context throughout conversations, ensuring that responses are relevant and coherent.

1.3.1. MERITS OF PROPOSED SYSTEM

Enhanced Natural Language Understanding: By integrating advanced natural language processing models, the chatbots can better comprehend and respond to user queries, including nuanced language and context. This leads to more accurate and relevant interactions, reducing frustration and improving customer satisfaction.

Personalization and Context Retention: The system's capability to retain context across interactions allows chatbots to provide personalized recommendations and solutions based on a user's history and preferences. This personalized approach enhances the overall customer experience and fosters stronger customer relationships.

Seamless Human-in-the-Loop Integration: Implementing a human-in-the-loop system ensures that complex or sensitive queries are smoothly escalated to human agents when necessary. This approach maintains service quality and resolution effectiveness, balancing automation with human expertise.

Continuous Learning and Improvement: Leveraging continuous learning mechanisms enables chatbots to evolve and improve over time. By analyzing real-time feedback and employing reinforcement learning techniques, the chatbots can adapt to new queries and optimize responses, enhancing overall performance and accuracy.

Multi-channel Support: The system's ability to operate across multiple channels such as web, mobile apps, and social media platforms ensures consistent service delivery regardless of the platform used by customers. This omni-channel support increases accessibility and convenience for users.

Ethical AI Guidelines and Security: Adherence to ethical AI principles ensures transparency, fairness, and accountability in chatbot interactions. Robust security measures protect customer data, maintaining trust and compliance with data privacy.

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE REVIEW:

Araujo (2018) emphasized that chatbots with human-like attributes are perceived more positively. This humanization can include giving the chatbot a name, enabling it to use natural language, and allowing it to exhibit emotions. Such features make the interaction more engaging and can lead to higher user satisfaction and compliance.

Qiu and Bombast (2009) explored the social relationship perspective, showing that anthropomorphic design in recommendation agents fosters trust and positive user attitudes, enhancing overall effectiveness.

Nass, Moon, and Carney (1999) studied politeness in human-computer interaction, revealing that users tend to apply social norms to computers, treating them with a degree of politeness usually reserved for human interactions. This finding underpins the importance of designing chatbots that can simulate polite and considerate interactions to improve user compliance.

Burger (1999) provided a comprehensive review of the FITD technique, a strategy that involves making a small initial request followed by a larger request. The study found that this technique is effective across various contexts, including digital interactions. Applying this to chatbots can enhance compliance by gradually increasing user commitment. The FITD technique leverages the principle of consistency, where individuals feel compelled to maintain consistency in their actions, leading them to comply with larger requests after agreeing to smaller ones.

Adam et al. (2019) discussed the role of anthropomorphism and personalized anchors in rob advisors, highlighting that these elements significantly influence user investment decisions. The findings suggest that personalized and human-like design cues can improve user satisfaction and engagement with AI systems.

The research by Adam et al. (2019) further supports the idea that users respond better to systems that provide personalized experiences, indicating a broader trend in AI-based systems towards customization and user-centred design.

Cialdini et al. (1975): Demonstrated the effectiveness of the reciprocity principle using the door-in-the-face technique. When participants were first presented with a large request and then a smaller one, they felt compelled to comply with the second request out of a sense of obligation.

Regan (1971): Found that participants were more likely to buy raffle tickets from someone who had given them a free soda earlier. This study highlights the impact of small gifts or favors on compliance behavior.

Freedman and Fraser (1966): Demonstrated the foot-in-the-door technique, where initial small commitments lead to larger ones. Participants who agreed to place a small sign in their yard were more likely to agree to a larger, more intrusive sign later on.

Cialdini et al. (1995): Showed that public commitments, such as signing a petition, increase the likelihood of future consistent behavior.

Bandura et al. (1967): Found that individuals are more likely to engage in behavior if they observe others doing it first. This concept is rooted in observational learning and social modeling.

Goldstein et al. (2008): Demonstrated that hotel guests were more likely to reuse towels when informed that the majority of other guests did do

Milgram (1963): Demonstrated the power of authority in obedience, where participants were willing to administer what they believed were harmful electric shocks to others under the instruction of an authority figure.

Chaiken (1979): Found that physically attractive communicators were more persuasive in
getting individuals to agree requests.

Worchel et al. (1975): Demonstrated that participants valued cookies more highly when they were told there were only a few left, compared to when there was an abundant supply.

Cialdini (2001): Found that highlighting the limited availability of products or opportunities increased their attractiveness and the urgency to act.

2.2 REQUIREMENT SPECIFICATIONS

2.2.1 HARDWARE REQUIREMENTS

The hardware specifications for the Ai based chatbots for customer service and user compliance project are outlined as follows:

- **CPU**: Multi-core processors (e.g., Intel Core i7/i9, AMD Ryzen 7/9).
- **Memory (RAM)**: At least 16 GB, but 32 GB or more is recommended for handling larger datasets and running virtual machines.
- **Storage**: SSDs with at least 512 GB capacity for quick data access.
- **GPU**: A dedicated GPU (e.g., NVIDIA RTX series) for local model training and testing

2.2.2 SOFTWARE REQUIREMENTS

The software components necessary for the project implementation include:

- **Operating system**: Linux distributions (e.g., Ubuntu, CentOS, Debian) are commonly used due to their stability, security, and support for a wide range of software.
- Windows, macOS, or Linux, depending on the developer's preference and compatibility with development tools.
- **Languages**: Python is the most widely used language for AI and machine learning due to its extensive libraries and community support. Other languages like JavaScript (Node.js), Java, and R can also be used depending on the project requirements.
- **Machine Learning/Deep Learning**: TensorFlow, PyTorch, Keras, Scikit-learn for building and training AI models.
- **Natural Language Processing (NLP):** NLTK, SpaCy, Hugging Face Transformers for processing and understanding human language.
- **Python:** A versatile language well-suited for data analysis, machine learning, and web development.
- **Django:** A popular Python web framework for building the backend of the system.
- **React:** A JavaScript library for building the frontend user interface.
- **TensorFlow or PyTorch**: Deep learning frameworks for training machine learning models.
- **Natural Language Toolkit (NLTK):** For natural language processing tasks.

2.2.3 FUNCTIONAL REQUIREMENTS

The functional requirements encompass the following key functionalities:

- **Natural Language Understanding (NLU**): The chatbot must accurately understand and interpret user inputs, including text, voice, and possibly even image-based queries.
- **Natural Language Generation (NLG**): The chatbot should generate human-like, contextually appropriate responses.
- **Multilingual Support**: If required, the chatbot should support multiple languages to cater to a diverse user base.
- **User Identification**: The chatbot should be able to identify and authenticate users through login credentials, social media accounts, or other secure methods.
- **Access Control**: Different levels of access should be provided based on user roles and permissions.
- **Session Management**: The chatbot must maintain the context of the conversation across multiple
- interactions and sessions.
- **Contextual Awareness**: The chatbot should remember user preferences and previous interactions to provide personalized responses.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

Choosing the correct architecture depends on what type of domain the chatbot will have. For example, you might ask a chatbot something and the chatbot replies to that. Maybe in mid-conversation, you leave the conversation, only to pick the conversation up later. Based on the type of chatbot you choose to build, the chatbot may or may not save the conversation history. For narrow domains a pattern matching architecture would be the ideal choice. However, for chatbots that deal with multiple domains or multiple services, broader domain. In these cases, sophisticated, state-of-the-art neural network architectures, such as Long Short-Term Memory (LSTMs) and reinforcement learning agents are your best bet. Due to the varying nature of chatbot usage, the architecture will change upon the unique needs of the chatbot.

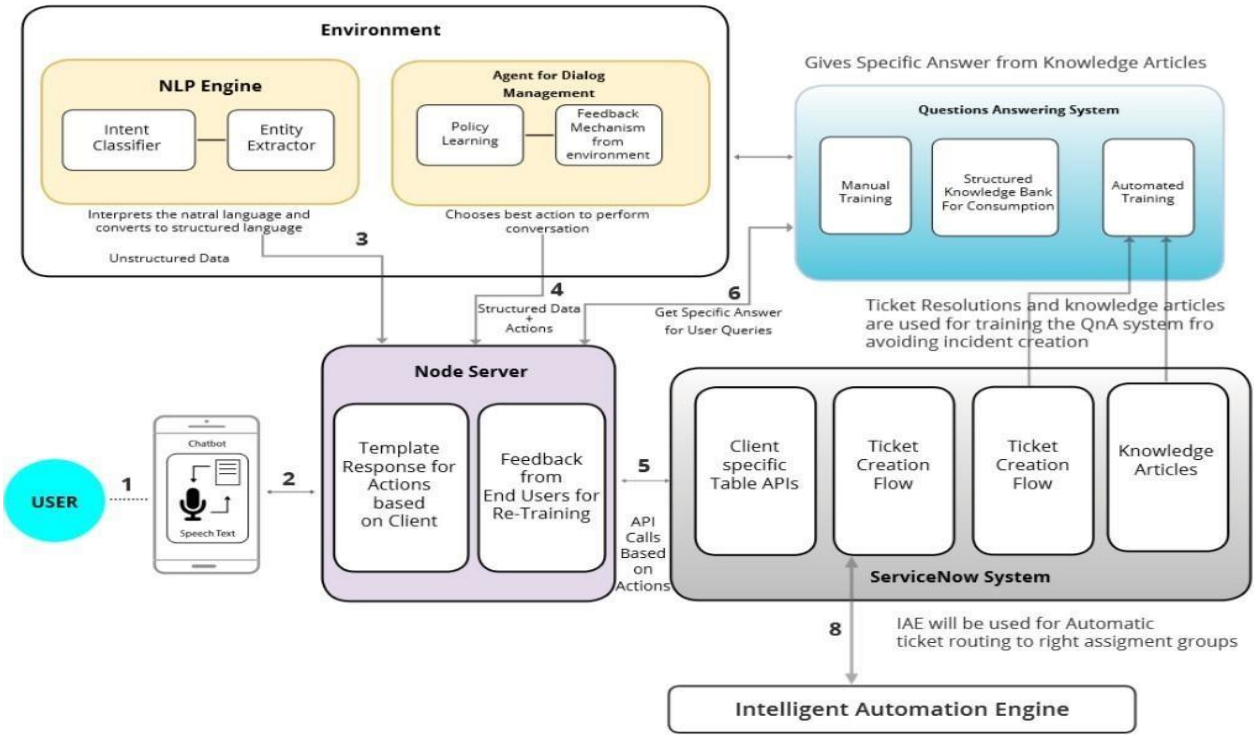


Figure 3.1: Conversational ChatBot model Architecture

Environment

This is where the core Natural Learning Process (NLP) engine and context interpretation happens.

NLP Engine

NLP Engine is the core component that interprets what users say at any given time and converts the language to structured inputs that system can further process. Since the chatbot is domain specific, it must support so many features. NLP engine contains advanced machine learning algorithms to identify the user's intent and further matches them to the list of available intents the bot supports

NLP Engine further has two components:

Intent Classifier: Intent classifier takes user's input identifies its meaning and relates back to one of the intents that the chatbot supports.

Entity Extractor: Entity extractor is what extracts key information from the user's query.

(Learn in detail about NLP engines and which NLP engine to choose)

Agent for Dialogue Management

It manages the actual context of the dialogue. For example, the user might say "He needs to order ice cream" and the bot might take the order. Then the user might say "Change it to coffee", here the user refers to the order he has placed earlier, the bot must correctly interpret this and make changes to the order he has placed earlier before confirming with the user.

Dialog management plugin enables us to do this.

Dialogue management further has following key plugins:

Feedback Mechanism: Here the agent takes the feedback from user time to time to learn if the bot is doing fine with the conversation and the user is satisfied with the bot's response. This reinforces the bot to learn from mistakes and corrects itself in future conversations. Policy Learning: Policy learning is a higher-level framework that teaches the bot to take more of happy paths during the conversation to improve overall end-user satisfaction.

Broadly it creates the network of happy paths and routes the conversation to end-user satisfaction.

The bot then tries to learn from the interactions and follows the interaction flow about the conversation it had with similar users in the past.

Question and Answer System

This is the key component in answering users' frequently asked questions. Q & A system interprets the question and responds with relevant answers from the knowledge base. It has the following components

Manual Training: Manual training involves the domain expert creating the list of frequently asked users queries and map its answers. This helps the bot quickly identify the answers to the most important questions.

Automated Training: Automated training involves submitting the company's documents like policy documents and other Q&A type of documents to the bot and ask it to train itself. The engine comes up with a list of question and answers from these documents. The bot then can answer with confidence.

Plugins/Components

Plugins offer chatbots solution APIs and other intelligent automation components for chatbots used for internal company use like HR management and field-worker chatbots.

Node Server / Traffic Server

The server that handles the traffic requests from users and routes them to appropriate components. The traffic server also routes the response from internal components back to the front-end systems.

Front-End Systems

Front-end systems can be any client-facing platforms. They can be the actual chatbot interfaces that reside in various platforms like:

Facebook Slack, Google, Hangouts, Skype for Business, Microsoft Teams

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

4.1 METHODOLOGY

Methodology refers to the systematic approach used to conduct research or development. For your proposed personalized learning system, a hybrid methodology combining agile development and research-driven design would be suitable.

Agile Development
Iterative and Incremental: Break down the project into smaller, manageable iterations (sprints).
Continuous Feedback: Gather feedback from users and stakeholders throughout the development process.
Flexibility: Adapt to changes and new requirements as needed.
Collaboration: Foster collaboration among team members and stakeholders. Research-Driven Design
User Research: Conduct user research to understand students' needs, preferences, and learning styles.
Literature Review: Review existing research on personalized learning, AI, and education. Expert Consultation: Seek input from experts in education, AI, and technology.

Preprocessing:

- **Data Collection:** Collected Data from LMS login, lecture notes, ppts
- **Data Cleaning:** Clean the data by removing irrelevant text, special characters.
- **Text Normalization:** Split sentences into words (tokenization).
- **Standardize Words:** Convert words to their base forms (e.g., "running" to "run") and remove common filler words (stop words).
- **Identify Key Elements:** Use techniques to recognize key subjects, course names, and relevant entities in the text.
- **Intent Label:** Label data with what users want (e.g., asking about a course or help with an assignment).
- **Build a Knowledge Base:** Create a structured database of educational resources, FAQs, and common responses.
- **Testing:** Set aside some data to test the chatbot's performance later.
- **Set Up the Environment:** AI frameworks and prepare the server or platform for deployment.
- **Continuous Improvement:** System gather user feedback and improve the chatbot over time.

Data Augmentation:

Data augmentation is an essential technique for enhancing the training dataset of an AI-based educational chatbot, ultimately improving its performance and robustness. One effective method is synonym replacement, where words in existing queries are substituted with their synonyms to create variations.
Paraphrasing involves rewording questions while retaining their meaning, such as transforming "What is the course about?" into "Can you tell me about the course?" Another approach is back translation, where a sentence is translated into another language and then back to the original language, generating new phrasing. Introducing noise, like small typos or grammatical errors, can simulate real user input and help the chatbot adapt to common mistakes.
Additionally, text expansion creates longer, more detailed versions of responses, enriching the dataset. Random insertion adds relevant words into sentences for further variation, while generating related questions helps create diverse queries around similar topics. Tailoring variations to specific subjects enhances contextual understanding, and creating complete dialogue scenarios can simulate realistic interactions.
Overall, these techniques not only increase the diversity of the dataset but also improve the chatbot's ability to handle various user expressions, leading to a more effective and versatile educational tool.

Output Prediction:

The integration of AI-based chatbots in the education system is set to revolutionize learning by offering a more.
Personalized, accessible, and efficient educational experience. These chatbots can adapt to individual student needs.
Providing tailored learning experiences that adjust to the student's pace and level of understanding.
This personalized approach can significantly enhance learning outcomes and bridge knowledge gaps.
Additionally, AI chatbots can offer 24/7 support, answering questions and assisting with assignments at any time
Any time, it increases student engagement and satisfaction. By automating routine tasks like grading.

Attendance, these chatbots free up teachers to focus on more meaningful, interactive teaching, thereby improving the quality of education.
Moreover, AI chatbots can deliver real-time feedback on assessments, allowing students to correct mistakes immediately, which accelerates the learning process. They also make education more inclusive by supporting students.
Supporting students with special needs and offering multilingual assistance, thus broadening the reach of education.
The data-driven insights provided by chatbots will enable educators to
identify areas where students struggle and tailor their teaching strategies accordingly. Additionally,
AI-powered chatbots have the potential to support lifelong learning and professional development, allowing individuals to continuously upskill in an ever-evolving job market.

4.2 Packages and Modules

In a AI Based Chatbot In Educational System has various packages, libraries, and modules are employed to facilitate data preprocessing, model development, training, evaluation, and deployment.

Transformers: The Hugging Face Transformers library is crucial for using Gemini BERT, providing pre-trained models and utilities for fine-tuning.

NLTK or SpaCy: For additional text processing tasks like tokenization, lemmatization, and named entity recognition.

Pandas: For data manipulation and analysis, especially useful for handling datasets and augmented data.

NumPy: For numerical operations, often used in conjunction with data handling.

Scikit-learn: For model evaluation and additional machine learning tasks, such as splitting data into training and testing sets.

PyTorch or TensorFlow: Depending on your preference, either of these frameworks can be used for fine-tuning the Gemini BERT model.

Flask or Fast API: For building the web application or API that will serve the chatbot.

Docker: For containerizing your application to simplify deployment and scaling.

Gemini BERT:

Gemini BERT, though less commonly known or defined, seems to reference a variant or extension of the original BERT architecture, potentially aimed at addressing specific issues or improving performance in certain NLP tasks. It may relate to research projects or implementations where BERT has been optimized or extended in unique ways

4.3 DATA SET

(https://akanksha.iare.ac.in/index.php?route=account/login)

When developing an AI-based chatbot for the educational system, sourcing the right datasets is crucial for its effectiveness. Potential sources we included are educational institutions learning management system , which can provide course materials such as syllabi, lecture notes, and textbooks. Open educational resources like online course platforms (e.g., Geeks for geek) offer valuable content and insights into common student questions. they present a wealth of information related can yield user-generated questions and answers across various subjects, This forms can help gather specific queries and needs from students and educators. Content libraries, including Wikipedia and Open Educational Resources (OER) repositories, further enrich the knowledge base. Finally, analysing discussions in educational social media groups can provide real-time insights into student challenges. By leveraging these diverse sources, we have created a comprehensive training dataset, ensuring the chatbot is well- equipped to support a wide range of student inquiries effect

Training: The training phase of the AI Based ChatBot In Educational System is done with training the model with 2 subjects python and machine learning notes fron lectures ,lms ppts,geeks for geeks

Testing: The testing phase of an AI-based educational chatbot ensures its functionality and user satisfaction through various strategies, including unit, integration, and user acceptance testing.
It involves engaging real users to gather feedback, validating API interactions, and assessing performance under load. Security and compliance checks safeguard user data and adhere to regulations. Continuous monitoring and documentation of results facilitate ongoing improvements, ultimately delivering a robust and effective tool for users.

4.4 SOURCE CODE

i. Requirements

requirements.txt

chat_bot_ext > requirements.txt

```
annotated-types==0.7.0
asgiref==3.8.1
cachetools==5.5.0
certifi==2024.8.30
charset-normalizer==3.3.2
colorama==0.4.6
Django==5.1
django-cors-headers==4.4.0
djangorestframework==3.15.2
google-ai-generativelanguage==0.6.6
google-api-core==2.19.2
google-api-python-client==2.143.0
google-auth==2.34.0
google-auth-httplib2==0.2.0
google-generativeai==0.7.2
googleapis-common-protos==1.65.0
grpcio==1.66.1
grpcio-status==1.62.3
httplib2==0.22.0
idna==3.8
proto-plus==1.24.0
protobuf==4.25.4
pyasn1==0.6.0
pyasn1_modules==0.4.0
pydantic==2.8.2
pydantic_core==2.20.1
pyparsing==3.1.4
requests==2.32.3
rsa==4.9
sqlparse==0.5.1
tqdm==4.66.5
typing_extensions==4.12.2
tzdata==2024.1
```

ii. Frontend -index page

index.html

C:\Users\likhitha_chowdary\Desktop\chat_bot_ext\frontend\public\index.html (preview)

```
<!doctype html>
<html lang="en">
<head>

    <meta charset="utf-8" />
    <link rel="icon" href="{% static 'favicon.ico' %}" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta name="description" content="Web site created using create-react-app" />
    <link rel="apple-touch-icon" href="{% static 'logo192.png' %}" />
    <link rel="stylesheet"
        href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <link rel="apple-touch-icon" href="{% static 'logo192.png' %}" />
    <link rel="manifest" href="{% static 'manifest.json' %}" />
    <title>React App</title>
    <script defer="defer" src="{% static 'static/js/main.7b6d7786.js' %}"></script>
    <link href="{% static 'static/css/main.dfd12ccb.css' %}" rel="stylesheet">
</head>
<body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
</body>
</html>
```

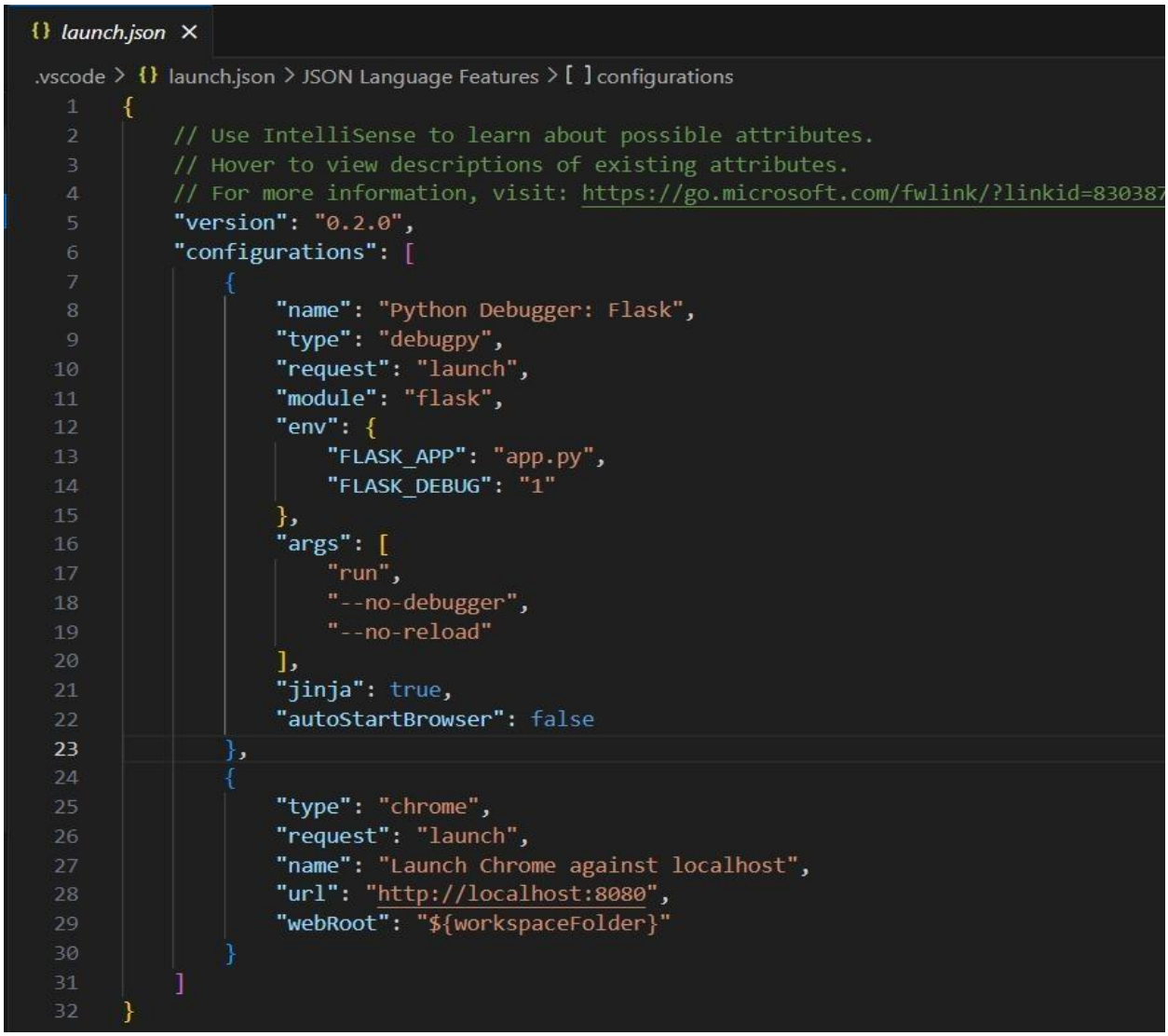
iii. Database

```
from django.db import models
from django.contrib.auth.models import AbstractBaseUser, BaseUserManag

class CustomUserManager(BaseUserManager):
    def create_user(self, email, username, password=None):
        if not email:
            raise ValueError('Users must have an email address')

        user = self.model(
            email=self.normalize_email(email),
            username=username,
        )

        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_superuser(self, email, username, password=None):
        user = self.create_user(
            email=email,
            username=username,
            password=password,
        )
        user.is_admin = True
        user.is_staff = True
        user.save(using=self._db)
        return user

class CustomUser(AbstractBaseUser):
    email = models.EmailField(unique=True)
    username = models.CharField(max_length=100, unique=True)
    date_joined = models.DateTimeField(auto_now_add=True)
    is_active = models.BooleanField(default=True)
```

iv. Database

```
class CustomUser(AbstractBaseUser):
    is_admin = models.BooleanField(default=False)
    is_staff = models.BooleanField(default=False)

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['username']

    objects = CustomUserManager()

    def _str_(self):
        return self.email

    def has_perm(self, perm, obj=None):
        return self.is_admin

    def has_module_perms(self, app_label):
        return self.is_admin




from django.db import models
from django.conf import settings

class ChatHistory(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE, related_name='chat_histories')
    chat_log = models.TextField()  # Store the chat history as JSON
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"ChatHistory of {self.user.email} at {self.created_at}"
```

v. API


launch.json
.vscode > launch.json > JSON Language Features > configurations
{
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
"version": "0.2.0",
"configurations": [
{
"name": "Python Debugger: Flask",
"type": "debugpy",
"request": "launch",
"module": "flask",
"env": {
"FLASK_APP": "app.py",
"FLASK_DEBUG": "1"
},
"args": [
"run",
"--no-debugger",
"--no-reload"
],
"jinja": true,
"autoStartBrowser": false
},
{
"type": "chrome",
"request": "launch",
"name": "Launch Chrome against localhost",
"url": "http://localhost:8080",
"webRoot": "${workspaceFolder}"
}
]
}

CHAPTER 5

RESULTS

Interpreting the results of an AI chatbot in the educational system involves analysing various metrics to understand its impact and effectiveness. Here are the result interpretations:

1. User Engagement

High interaction rates and prolonged session durations indicate that students find the chatbot engaging and useful. If engagement drops, it may suggest that the chatbot needs to be more interactive or relevant to student needs.

2. Learning Outcomes

Improvements in test scores and higher completion rates for learning materials suggest that the chatbot effectively supports academic success. Comparing these metrics before and after the chatbot's implementation can provide a clearer picture of its contribution to learning.

3. User Satisfaction

Feedback collected through surveys and ratings can reveal students' perceptions of the chatbot. High satisfaction scores typically indicate that the chatbot meets user expectations, while low scores might highlight areas for improvement.

4. Response Accuracy

Evaluating the correctness of the chatbot's responses is crucial. High accuracy rates demonstrate reliable support, while frequent inaccuracies may necessitate updates to its knowledge base or training data.

5. Accessibility and Inclusivity

Analyzing usage statistics among diverse student populations can identify whether the chatbot is effectively reaching all users. Ensuring it accommodates various learning styles and abilities is essential for fostering an inclusive learning environment.

6. **Comparison with Traditional Methods**

Assessing how the chatbot performs relative to traditional educational resources—such as tutors or textbooks—can provide insights into its efficiency. Metrics like time saved in finding information can highlight the chatbot's advantages.

AI chat systems have transformed the education sector, offering new ways for students and educators to engage with knowledge. These systems enhance personalized learning by providing instant feedback and allowing students to learn at their own pace. They also assist educators in monitoring progress and addressing individual learning needs. However, despite the advantages, AI chatbots should complement rather than replace human teachers, as human interaction remains essential for a holistic educational experience. Looking ahead, the future of AI in education is promising, with systems becoming more personalized and capable of delivering tailored learning experiences based on student progress.

Advancements in natural language processing (NLP) will make these systems more contextaware, offering accurate responses even for complex subjects. AI chat systems are also likely to expand multilingual support and accessibility, reaching underserved populations and bridging educational gaps. Moreover, they may evolve into advanced virtual teaching assistants, handling administrative tasks and providing emotional support to students. The integration of AI with technologies like augmented reality (AR) and virtual reality (VR) could create immersive learning experiences, further enhancing education. As AI continues to grow in education, ethical considerations around bias, data privacy, and inclusivity will be vital for ensuring its responsible and effective use. Ultimately, AI chat systems have the potential to revolutionize education, but their development must prioritize ethical, balanced, and inclusive approaches.

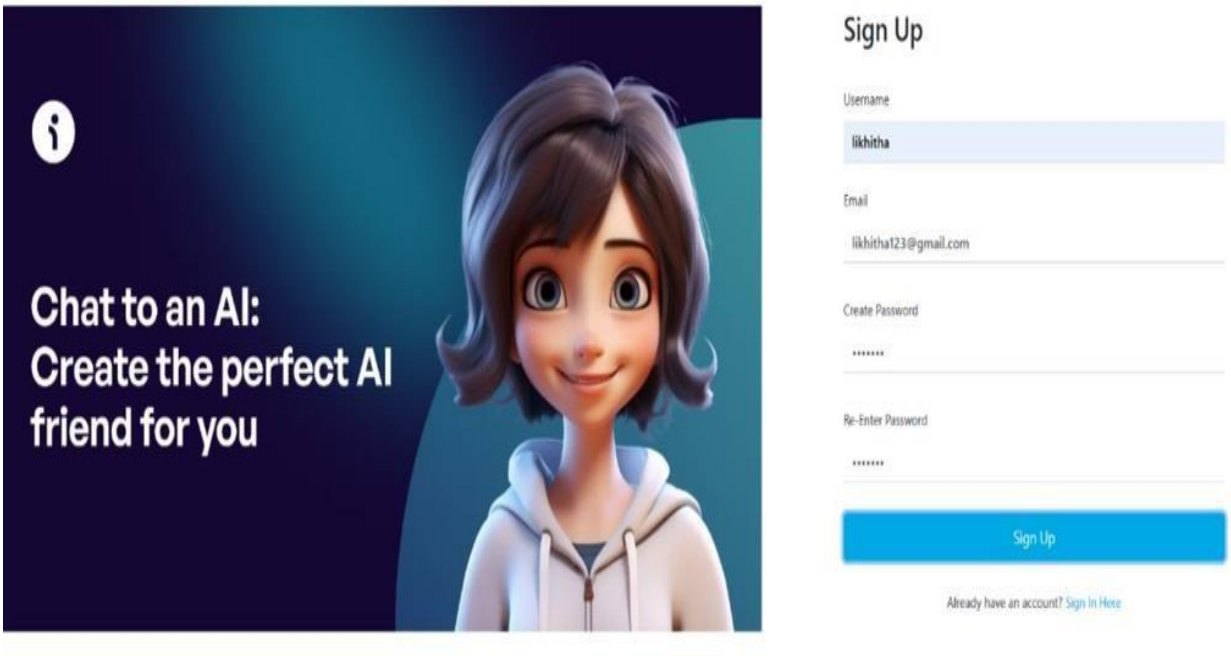


Fig.5.1.login page of chatbot

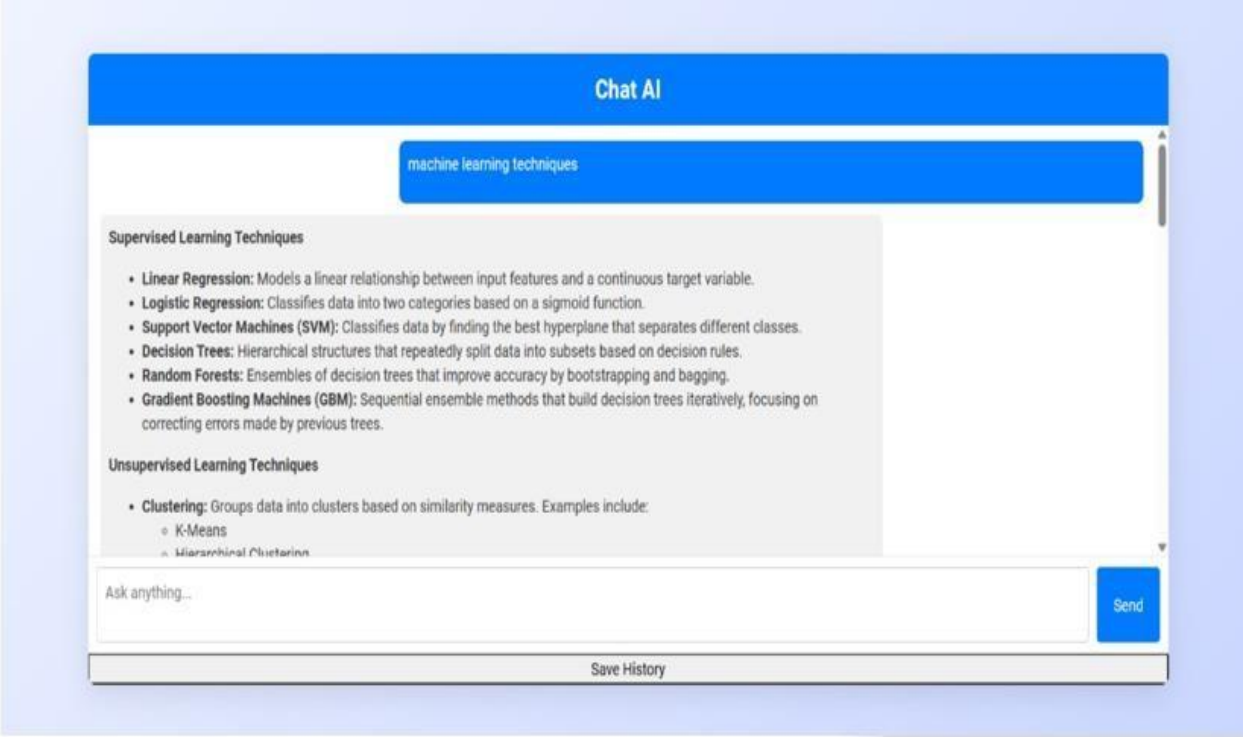


Fig.5.2.output of the chatbot

CHAPTER 6

CONCLUSION

- The development of a chatbot akin to ChatGPT for college students represents a significant advancement in the realm of educational technology. This project underscores the transformative potential of integrating sophisticated AI-driven tools within academic institutions to enhance the student experience. By providing 24/7 access to personalized, real-time support, the chatbot addresses key challenges in traditional student services, such as limited availability, slow response times, and fragmented information sources.
- Through its ability to handle a wide range of inquiries—ranging from academic guidance.
- The chatbot aims to streamline support processes and reduce the burden on staff. Its integration with institutional systems ensures that students receive accurate and contextually relevant information, while its continuous learning capabilities enable it to adapt and improve over time.
- Despite the promising benefits, it addresses potential drawbacks such as contextual limitations, privacy concerns, and the risk of over-reliance. Ongoing maintenance, rigorous testing, and user feedback will be essential in refining the chatbot's performance and ensuring it meets the diverse needs of the student body effectively.

REFERENCES:

Adam, M., Toutaoui, J., Pfeuffer, N., & Hinz, O. (2019). Investment decisions with roboadvisors: The role of anthropomorphism and personalized anchors in recommendations. In: Proceedings of the 27th European Conference on Information Systems (ECIS). Sweden:

Stockholm & Uppsala.

Agarwal, R., & Prasad, J. (1998). A conceptual and operational definition of personal innovativeness in the domain of information technology. Information Systems Research, 9(2), 204–215

Aggarwal, P., Vaidyanathan, R., & Rochford, L. (2007). The wretched refuse of a teeming shore? A critical examination of the quality of undergraduate marketing students. Journal of Marketing Education, 29(3), 223–233.

Benlian, A., Klumpe, J., & Hinz, O. (2019). Mitigating the intrusive effects of smart home assistants by using anthropomorphic design features: A multimethod investigation.

Information Systems Journal.

Burger, J. M. (1986). Increasing compliance by improving the deal: The that's-not-all technique.Journal of Personality and Social Psychology, 51(2), 277.

Chaiken, S. (1980). Heuristic versus systematic information processing and the use of source versus message cues in persuasion. Journal of Personality and Social Psychology, 39(5), 752.