Krmu chat

Chatbot for K.R MANGALAM UNIVERSITY

**STUDENTS DETAILS :**

# Nishant yadav Neeraj joshi

**ROLL : 2401010090 ROLL : 2401010044**

**SECTION – C SECTION - C**

**BTECH CSE CORE BTECH CSE CORE**

# Acknowledgement

We express our heartfelt gratitude to K.R. Mangalam University for providing us with the opportunity to work on this project. We thank our professors and mentors for their guidance and support throughout the development of this intelligent chatbot. Our sincere thanks to our peers and family for their constant encouragement.

# Abstract

In the digital age, students and applicants expect immediate responses to queries regarding academics, admissions, and campus services. The KRMU Chatbot is an AI-powered conversational assistant integrated into the university website to provide round-the-clock support. Built using Dialogflow and Microsoft Bot Framework, and powered by a backend in Node.js and MySQL, the chatbot delivers personalized and real-time responses. It streamlines communication, minimizes administrative workload, and enhances the user experience for both current and prospective students.

# Introduction

Traditional student support services often involve delays and repetitive manual tasks. The KRMU Chatbot automates these interactions by providing intelligent, conversational assistance. The system integrates with the university’s digital ecosystem and supports queries related to admissions, course offerings, faculty, events, and more. Its AI capabilities ensure accurate and contextual responses, significantly improving the university's digital accessibility and responsiveness.

# Problem Statement

Universities face several challenges in handling high volumes of student queries, especially during admission seasons:

* Long wait times for responses
* Limited office hours
* Repetitive manual work for administrative staff
* Lack of real-time, personalized support

There is a need for a scalable, automated system that offers reliable and instant communication 24/7.

# Objectives

The objectives of the **krmuchat** project are:

* To develop a 24/7 AI-powered chatbot for K.R. Mangalam University’s official website.
* To automate repetitive administrative queries and improve communication.
* To provide quick access to academic and campus-related information.
* To ensure scalable integration with existing digital infrastructure.
* To enhance user experience for both students and applicants.

# Literature Survey

Research into AI chatbots in education highlights their effectiveness in:

* Reducing administrative overhead
* Providing consistent and timely responses
* Supporting hybrid/online learning environments

Platforms like Dialogflow and Microsoft Bot Framework are widely used for building intelligent conversational agents due to their NLP capabilities and integration support. Studies suggest that chatbots improve student engagement and streamline university operations.

* **Data-Driven Education**: The use of digital data enhances decision-making and monitoring student progress.
* **Role-Based Systems**: Systems that provide distinct access to different types of users reduce information overload and maintain security.
* **Importance of Automation**: Automating repetitive tasks saves time, reduces errors, and improves operational efficiency.

Existing solutions often suffer from complexity, poor UI/UX design, high costs, and a lack of modularity. Krmuchat is designed to overcome these limitations by providing a lightweight, highly responsive, and modular system built with modern web technologies.

# System Analysis

## Functional Requirements

## Handle queries on admissions, faculty, courses, and campus life.

## 24/7 real-time chat functionality.

## Role-specific query handling (applicants, current students).

## Integration with databases for dynamic content.

## Non-Functional Requirements

## High availability and uptime

## Scalable backend infrastructure

## Secure data handling and privacy compliance

## Feasibility Study

## Technical: Dialogflow and Microsoft Bot Framework provide robust NLP and integration support.

## Operational: Minimal staff training required; automation reduces human intervention.

## Economic: Cloud deployment reduces infrastructure cost.

## Feasibility Study

* + - **Technical Feasibility**: React.js, Node.js, Express, and MongoDB provide an ideal stack for building dynamic, secure, and scalable applications.
    - **Operational Feasibility**: EduTrack is designed to easily integrate into existing school operations without disrupting workflows.
    - **Economic Feasibility**: Being a web-based system hosted on platforms like Netlify reduces the infrastructure costs significantly.

# System Design

## Architecture

## Frontend: Web interface (university website)

## Bot Engine: Dialogflow/Microsoft Bot Framework

## Backend: Node.js APIs

## Database: MySQL (for dynamic responses and data lookup)

## Hosting: Google Cloud Platform

## Module Descriptions

## Query Processor: Identifies user intent and extracts parameters.

## Response Manager: Fetches static/dynamic content based on query.

## Admin Panel: For managing FAQs and chatbot training data.

## Database Interface: Communicates with MySQL for real-time data retrieval

## Module Descriptions

* + - **Admin Dashboard**:
      * Manage users and roles
      * Monitor system activities
      * Publish announcements
    - **Teacher Dashboard**:
      * Upload and manage assignments
      * Mark attendance
      * Enter and view student results
    - **Student Dashboard**:
      * View assignments, results, attendance records
      * Pay and track fees
    - **Parent Dashboard**:
      * Monitor student academic and fee status
      * Communicate with teachers
    - **Chatbot**:
      * Provide instant answers to FAQs
      * Navigate users through different modules

# Technologies Used

1. **Bot Platform**: Dialogflow, Microsoft Bot Framework
2. **Backend**: Node.js, Express.js
3. **Database**: MySQL
4. **Cloud Services**: Google Cloud Platform
5. **Tools**: GitHub, Postman, Visual Studio Code
   * Git and GitHub (Version Control)

# Database Design

1. **Tables**:
   1. FAQs (id, question, answer)
   2. Logs (session\_id, query, response, timestamp)
   3. Courses, Faculty, Events (used for dynamic replies)

# Implementation Details

1. Chatbot intents and entities designed in Dialogflow.
2. Node.js middleware for dynamic data fetching from MySQL.
3. Secure REST APIs for communication.
4. Integrated into the university website using an iframe or web widget.

# Testing

1. **Unit Testing**: On intents, responses, and API endpoints.
2. **Integration Testing**: Chat flow with backend APIs.
3. **End-to-End Testing**: User queries to final response delivery.
4. **Security Testing**: SQL injection, input validation.

Tools used: Postman, Dialogflow Simulator, MySQL Workbench

# Results and Discussion

1.  24/7 automated assistance for 100+ FAQs
2.  85% reduction in repetitive admin queries
3.  Average response time: <1 second
4.  Positive feedback from test users and staff

# Future Scope

**krmuchat** can be expanded in the future by incorporating:

** Multi-language support (Hindi, regional languages)**

** Mobile app integration**

** Voice assistant support**

** Machine learning–based intent prediction**

** Integration with student portal for personalized data**

# Conclusion

1. The KRMU Chatbot project showcases how AI can be leveraged to modernize university operations. It simplifies interactions, boosts efficiency, and provides scalable, smart assistance to students and applicants alike.

# References

1.  Dialogflow Official Docs
2.  Microsoft Bot Framework Documentation
3.  Node.js and Express.js Docs
4.  MySQL Database Design Guides
5.  Google Cloud Deployment Docs