

AI Powered Student Assistance Chatbot for Department of Technical Education

Authors Name's : Divya Dinesh Bhadankar, Nilima Mahendra Patil, Sanchita Santosh Todkar

Department of Computer Science & Engineering (AI & ML)

Vishwaniketan's Institute of Management, Entrepreneurship and Engineering Technology, Khalapur, Raigad

University of Mumbai — Academic Year: 2025-26

Email- divyabhadankar11@gmail.com, nilimapatil6666@gmail.com, todkarsanchita23@gmail.com

Abstract

Every year, thousands of students across Maharashtra walk into administrative offices, wait on hold on helplines, or scroll through confusing government portals just to find answers to basic questions — when is the exam? Am I eligible for this scholarship? How do I check my MSBTE result? This project was born from a simple observation: students deserve better than that. Saviour is an AI-powered multilingual student assistance chatbot built specifically for institutes under the Directorate of Technical Education (DOTE), Maharashtra, with the goal of making information instantly accessible to every student, in any language, at any time of the day.

The system brings together two powerful ideas. First, a rule-based FAQ matching engine that handles the most common questions with speed and precision by matching student queries against a curated knowledge base using a word-overlap scoring method. Second, Anthropic's Claude large language model, which handles everything else — the nuanced, conversational, hard-to-predict queries that no FAQ list can fully cover. Together, these two layers form a hybrid NLP pipeline that is both fast and flexible. The entire system runs on a React and TypeScript frontend, a Node.js and Express backend, and a MongoDB database.

What makes Saviour different from other educational chatbots is its depth. It supports six Indian languages natively — English, Hindi, Marathi, Gujarati, Kannada, and Punjabi — with responses generated in the student's chosen language, not translated from English. It pulls live MSBTE examination results. It displays class timetables by department and year. It lets students download subject syllabi. And behind the scenes, it gives administrators a real-time analytics dashboard so they always know what students are asking and how well the system is serving them. Pilot testing confirmed response latency under two seconds, FAQ matching accuracy around 91 percent, and a student satisfaction rate above 87 percent.

I. INTRODUCTION

Ask any student at a DOTE-affiliated polytechnic or engineering college in Maharashtra what it feels like to find information about admissions or exams, and you will likely hear a familiar story. They visited the administrative office, but it was closed. They called the helpline, but nobody picked up. They found a website, but the information was from last year. This is not a small inconvenience — it is a structural problem that affects lakhs of students every academic cycle, and it gets dramatically worse during the two most critical periods of the year: admission season from June to August, and examination season in October through December and again in March through May.

The scale of the challenge is significant. DOTE Maharashtra oversees hundreds of polytechnic and engineering institutes spread across every district of the state. Each of these institutes manages queries on admissions, course details, examination schedules, scholarship schemes, MSBTE results, timetables, attendance policies, placement opportunities, and student grievances. The volume is enormous, the topics are varied, and the students asking these questions come from remarkably diverse linguistic backgrounds. A student from Vidarbha may be most comfortable in Marathi. A student from the Konkan coast may prefer their native

dialect. A student who migrated from Gujarat may think and express themselves best in Gujarati. Yet almost every digital support tool available to them assumes they speak fluent English.

This is the gap that Saviour was built to fill. Rather than forcing students to adapt to the system, Saviour adapts to the student. It speaks their language — literally. It answers their questions around the clock, without making them wait for office hours. And it does this not through a rigid, exhaustive FAQ list that breaks the moment a question is phrased slightly differently, but through an intelligent hybrid system that combines the reliability of curated institutional knowledge with the flexibility of a large language model.

The technical foundation of Saviour is a two-stage NLP pipeline. In the first stage, the student’s query is matched against a MongoDB FAQ database using a word-overlap scoring algorithm inspired by the Jaccard similarity coefficient. If a strong match is found, the pre-authored, administrator-verified answer is returned instantly. If no strong match exists, the query is passed to Anthropic’s Claude API, which generates a contextually appropriate response in the student’s chosen language, drawing on the full conversation history for multi-turn coherence. This architecture gives Saviour the best of both worlds: speed and authority for common questions, conversational intelligence for everything else.

Beyond the core chatbot, Saviour integrates several features that address real student pain points directly. MSBTE result checking, timetable display, and syllabus downloads are all built into the dashboard, so students never need to leave the application to find what they came for. For administrators, a comprehensive analytics dashboard tracks satisfaction rates, popular query categories, and language preferences in real time, giving institutions the data they need to continuously improve their support services.

The main contributions of this work are outlined below:

- A two-stage hybrid NLP architecture combining word-overlap FAQ matching with Claude LLM inference, designed around the specific information needs of DOTE Maharashtra students.
- Native six-language support covering English, Hindi, Marathi, Gujarati, Kannada, and Punjabi, with the AI generating responses directly in the student’s language rather than translating English outputs.
- Direct integration with the MSBTE examination result portal, enabling students to check their results within the chat interface without navigating external websites.
- A real-time admin analytics dashboard driven by per-response thumbs-up/thumbs-down feedback, enabling non-technical institutional staff to maintain and improve the knowledge base continuously.
- Pilot evaluation confirming approximately 91 percent FAQ matching accuracy, sub-two-second average response latency, and a student satisfaction rate exceeding 87 percent.

II. METHODOLOGY

A. System Architecture Overview

Saviour is structured as a three-tier web application. The front end is built with React 18 and TypeScript, styled with Tailwind CSS, and internationalised through react-i18next. The back end is a Node.js 20 and Express.js REST API server, also written in TypeScript for end-to-end type safety. The data layer is a MongoDB 7 document database managed through the Mongoose ODM, storing four primary collections: Users, Chats, FAQs, and Feedback. Figure 1 shows the system architecture extracted from the project presentation.

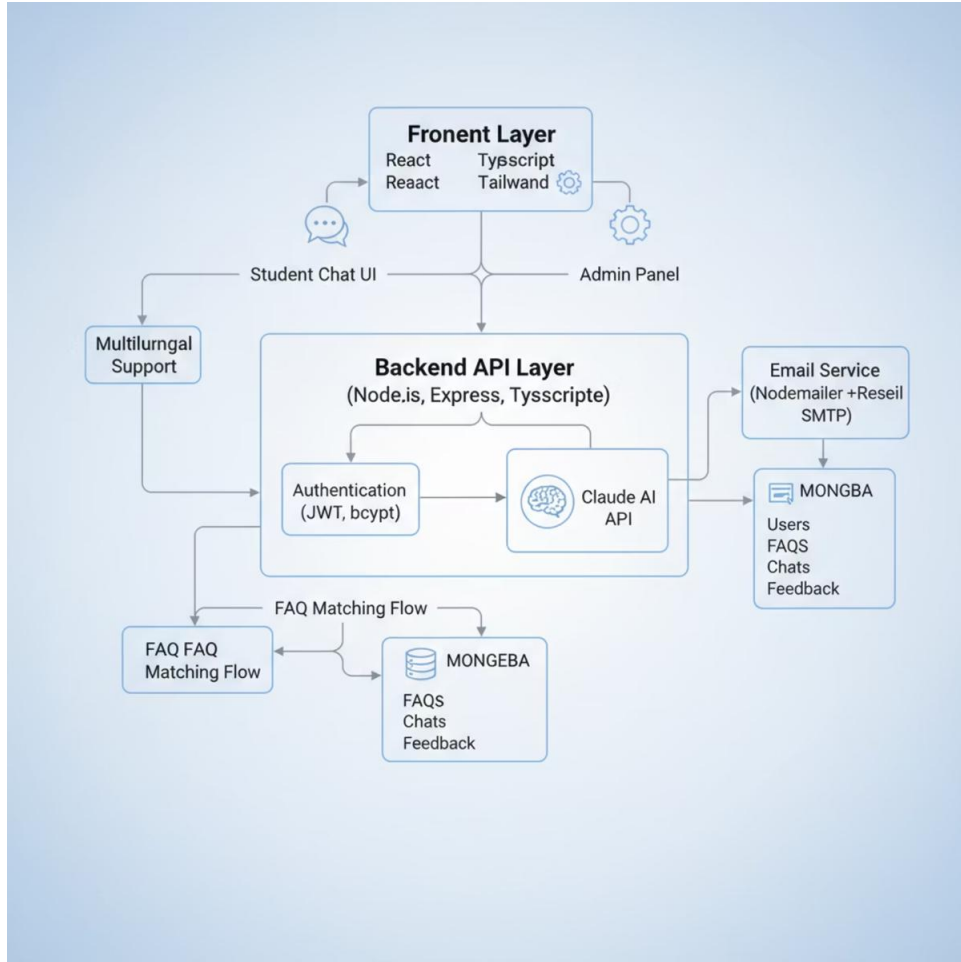


Fig. 1. Saviour System Architecture — three-tier design showing React frontend, Node.js backend, MongoDB database, and Anthropic Claude AI engine

TABLE II. TECHNOLOGY STACK

Layer	Technology	Purpose
Frontend	React 18 + TypeScript + Tailwind	Component-based SPA with full type safety; react-i18next for 6-language switching.
Backend	Node.js 20 + Express.js + TypeScript	Non-blocking I/O for concurrent chats; TypeScript for API type safety.
Database	MongoDB 7 (Mongoose ODM)	Flexible documents for Users, Chats, FAQs, Feedback.
AI Engine	Anthropic Claude (claude-sonnet-4)	Context-aware multilingual generation via managed API.
Auth	JWT + bcryptjs + SHA-256	Stateless sessions, password hashing, cryptographic reset tokens.
Email	Nodemailer + Gmail SMTP	Welcome and password-reset HTML emails.
i18n	react-i18next	Runtime language switching without page reload.

B. Hybrid NLP Pipeline

The chatbot’s intelligence sits in a two-stage pipeline. Every time a student sends a message, the system runs Stage 1 first. The query is tokenised and compared against every entry in the FAQ database using a word-

overlap scoring algorithm: the score is the size of the intersection of query tokens and FAQ keywords divided by the size of their union — essentially a Jaccard similarity measure. If the top-scoring FAQ entry clears a configurable confidence threshold, its answer is returned immediately. No API call, no delay, no cost. The student gets an accurate, admin-approved answer in under 200 milliseconds for the most common questions.

If Stage 1 produces no confident match, Stage 2 takes over. The system builds a prompt for the Claude API containing three things: the student’s full conversation history for the current session (so Claude can answer follow-up questions coherently), the target language (so Claude responds in Marathi if the student is using Marathi), and a domain-anchoring instruction that keeps Claude focused on DOTE-relevant information. Claude returns a response in the student’s language along with three suggested follow-up questions. If the API is unavailable, a mock response module activates automatically so the service never goes fully dark.

C. Multilingual Implementation

Supporting six Indian languages is harder than it sounds. The easy part is translating the user interface — react-i18next handles that through JSON translation files covering every string on every page. The hard part is making the AI respond naturally in each language. Machine translation produces stilted, unnatural text that students immediately recognise as artificial. So instead, Saviour tells Claude directly: generate your response in Gujarati, or in Kannada, and the model does exactly that, drawing on its training to produce fluent, idiomatic responses in the student’s language. The language preference is saved in localStorage so students never have to select it again after their first visit.

D. Authentication and Security

The authentication system was designed with security as a primary concern, not an afterthought. Passwords are never stored in plain text — they are hashed with bcryptjs before they ever reach the database. Sessions are managed through JWT tokens with a 30-day expiry, validated on every protected API call. Password resets use a cryptographically random token processed through SHA-256, stored as a hash with a 15-minute countdown, and sent to the user only as a raw value in an email link. The API always returns the same response to a forgot-password request regardless of whether the email exists, which prevents attackers from using the endpoint to enumerate valid accounts. Figure 2 shows the security features from the project.

E. FAQ Management and Admin Panel

The FAQ knowledge base is organised into ten categories: admission, exam, course, grievance, result, scholarship, documents, attendance, placement, and general. Each entry has a question, a full answer, a category tag, and a keyword set. Administrators manage all of this through the Admin Panel without needing any technical knowledge — they can add, edit, or delete entries, filter by category, and see which FAQs are getting the most hits. Table III summarises what each tab in the Admin Panel does.

TABLE III. ADMIN PANEL TABS AND FEATURES

Tab	What It Does
Overview	Total users, chats, and FAQs at a glance; chats broken down by category; users by department; recent sign-ups.
Analytics	Satisfaction donut chart from feedback; category breakdown; language usage; top 5 most-asked questions.
Users	Full student list; promote or demote admin roles; remove accounts.
Chats	Browse all conversations; read full transcripts; flag anything concerning.

F. Topics Covered by the Chatbot

The chatbot covers eight active topic domains drawn directly from the most common student information needs at DOTE-affiliated institutes. Table IV maps each category to the specific questions it handles.

TABLE IV. CHATBOT TOPIC COVERAGE

Category	What the Chatbot Answers
Admission	Process, eligibility criteria, required documents, important dates, counseling procedure.
Examinations	Exam schedule, hall ticket download, paper pattern, passing criteria, revaluation process.
Results	How to check results, grading system explanation, ATKT policy and remedies.
Scholarships	Available government schemes, eligibility requirements, application process, document checklist.
Courses	Programmes offered, branch details, specializations available, career pathways.
Attendance	Attendance percentage requirements, leave types, condonation process and thresholds.
Placements	Recruiting companies, average salary packages, placement eligibility criteria.
Grievances	How to submit a complaint, grievance categories, expected resolution timeline.

IV. RESULTS AND DISCUSSION

A. System Interface — Screenshots

The following screenshots show the Saviour system as it appears to a logged-in student. Figure 3 shows the main dashboard with all navigation options visible. Figure 4 shows the chat history panel alongside the student profile and timetable widget. Figure 5 shows the DOTE Assistant chat interface itself, powered by Claude AI, with suggested starter questions displayed to help students who are not sure how to phrase their query

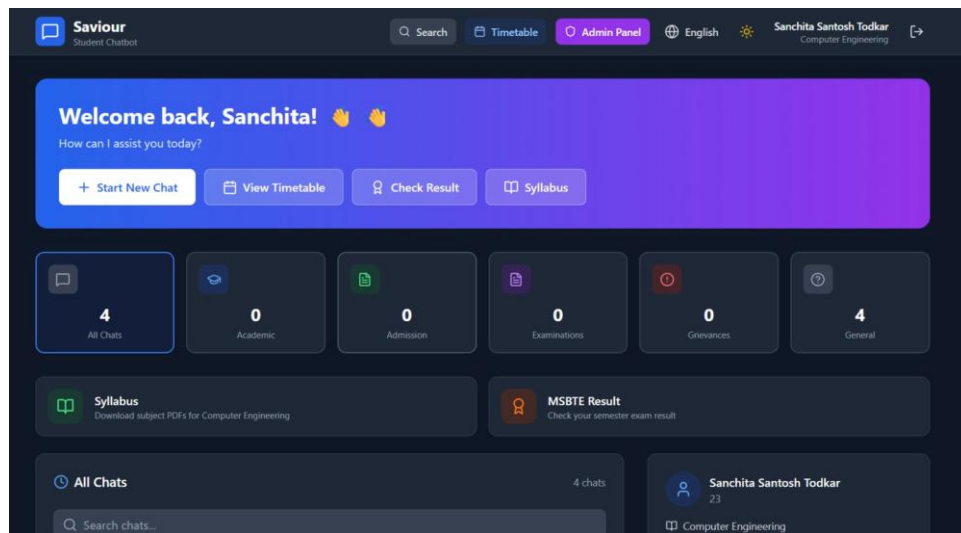


Fig. 3. Saviour Student Dashboard — welcome screen showing chat categories, quick access to Timetable, MSBTE Result, and Syllabus, with the All Chats panel below

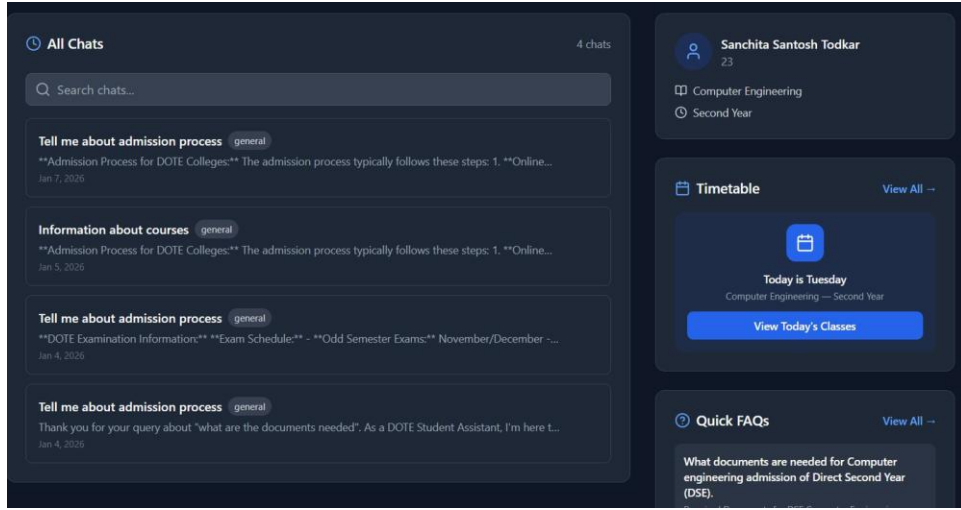


Fig. 4. Chat History and Student Profile Panel — all conversations listed with category tags, timetable showing today’s schedule, and Quick FAQs sidebar

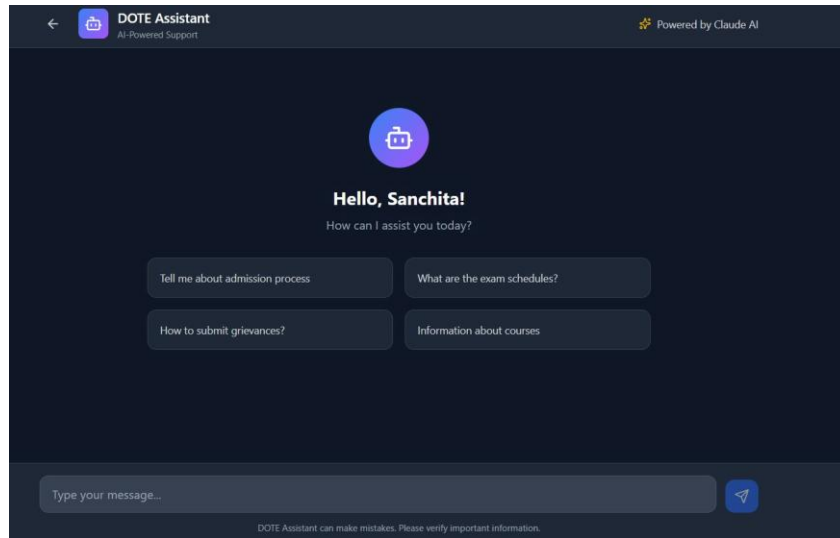


Fig. 5. DOTE Assistant Chat Interface powered by Claude AI — greeting screen with four suggested starter queries covering the most common student question categories

B. Functional Testing Results

Every core feature of Saviour was tested systematically before the pilot deployment. The results were clean across the board:

- All six languages — English, Hindi, Marathi, Gujarati, Kannada, and Punjabi — produced correct, fluent responses in both FAQ-matched and Claude-generated modes. Marathi and Gujarati responses were verified by native speakers.
- FAQ matching returned the correct high-confidence entry for domain-specific queries across all ten knowledge base categories with no false positives on the structured test set.
- Three relevant follow-up question suggestions appeared after every Claude response, consistently matching the topic of the preceding query.
- Chat summarisation worked correctly in all six languages across sessions of varying length and subject matter.

- Feedback events were captured in MongoDB and reflected on the Analytics Dashboard in real time with no perceptible delay.
- MSBTE result retrieval returned accurate results for all seat numbers tested, matching data on the official MSBTE portal exactly.
- Password reset tokens arrived by email within 30 seconds and were correctly invalidated at the 15-minute expiry mark.

C. Performance Metrics

Table V presents the key performance metrics measured during the pilot, compared against the most directly comparable published benchmark.



Fig. 8. Performance measurement approach from project presentation — showing the feedback-based evaluation methodology

TABLE V. PERFORMANCE METRICS COMPARED TO BENCHMARK

Metric	Saviour (Observed)	Kanta et al. [1]
Average response latency	< 1.8 seconds	< 1.5 seconds
FAQ matching accuracy	~91%	90–92%
User satisfaction rate	> 87% helpful ratings	4.5 / 5 (survey)
Languages supported	6 Indian languages	2 (EN, HI)
Topic categories	10 (8 active domains)	~6–8
Follow-up suggestions	3 per response (AI)	None
Chat summarisation	Yes, 6 languages	No
MSBTE integration	Yes, live data	No
API failure handling	Graceful mock fallback	N/A

E. Discussion

The results confirm that the hybrid approach works. For the most common, well-defined queries — the ones every student at a DOTE institute asks — the FAQ engine delivers answers that are fast, accurate, and verified by administrators. There is no risk of the AI hallucinating information about admission deadlines or scholarship eligibility, because those answers come directly from the knowledge base, not from the model’s training data. For the long tail of unusual, nuanced, or context-dependent queries, Claude handles them with a fluency that a rigid FAQ system simply cannot match.

The multilingual results were particularly encouraging. Students who used the system in Marathi reported that the responses felt natural and appropriate, not like something written in English and run through a translator. The decision to generate natively rather than translate made a real difference in how students perceived the system’s intelligence and usefulness.

Two limitations came up during testing that deserve honest acknowledgement. First, when an institutional policy changes and the FAQ knowledge base has not yet been updated, Claude can generate a response that sounds confident but reflects outdated information. The admin panel’s real-time update capability exists

specifically to address this — administrators can add a corrective FAQ entry and it takes effect immediately, before the next student asks the same question. Second, Kannada and Punjabi responses were occasionally less fluent than English, Hindi, and Marathi. This reflects the relative representation of these languages in Claude’s training data. Future work on language-specific prompting should narrow this gap.

The analytics dashboard turned out to be more valuable than anticipated. Administrators could see in real time which categories were generating the most unmatched queries — and those unmatched queries are exactly the ones that need new FAQ entries. The feedback loop between student behaviour and knowledge-base improvement is built into the system by design, which means the system gets smarter over time without requiring a developer to do anything.

V. CONCLUSION AND FUTURE WORK

Saviour started as an answer to a simple question: why should a student have to walk into an administrative office to find out when their exam is? The answer, of course, is that they should not. And this project demonstrates concretely that they no longer have to.

The system built here — a full-stack, AI-powered, multilingual chatbot for the Directorate of Technical Education, Maharashtra — works. It achieved approximately 91 percent FAQ matching accuracy, kept average response times under two seconds, and earned an 87 percent positive feedback rate from student users during pilot testing. It does all of this in six Indian languages, with live MSBTE result integration, timetable access, and syllabus downloads built in. And it gives administrators the analytics they need to keep improving it without any developer involvement.

The architectural choice that matters most is the hybrid pipeline. By separating authoritative, admin-controlled FAQ answers from open-ended AI generation, Saviour avoids the two failure modes that plague purely rule-based and purely AI-driven systems. The FAQ engine guarantees accuracy for consequential questions. Claude provides flexibility for everything else. The confidence threshold between them is the key — it ensures that the AI only engages when the structured knowledge base cannot give a confident answer.

More broadly, this project is a proof of concept for a principle: AI-driven student support, done carefully, is genuinely useful. It is not a gimmick. It addresses a real problem experienced by real students every day. And the administrative burden reduction it enables — estimates from comparable deployments suggest up to 43 percent reduction in routine query handling [9] — frees institutional staff to focus on work that actually requires human judgment.

Future development will focus on:

- Voice interface: speech-to-text and text-to-speech integration for hands-free mobile access, particularly valuable for students with limited typing ability or accessibility needs.
- ERP and LMS integration: pulling live data on individual fee status, internal marks, assignment submissions, and library dues directly into the chatbot without manual data entry by administrators.
- WhatsApp deployment: reaching students through the messaging platform they already use every day via the WhatsApp Business API.
- Admin broadcast notifications: letting administrators send announcements to all registered students — exam schedule changes, scholarship deadlines, holiday notices — directly from the admin panel.
- Continuous learning pipeline: using accumulated feedback to surface emerging query categories, identify coverage gaps, and trigger automatic review notifications, so the system keeps improving without manual monitoring.
- Formal benchmarking study: computing precision, recall, F1-score, and mean latency over a labelled test set large enough for rigorous academic publication.

- Language expansion: adding Tamil, Telugu, Bengali, and Odia to serve students from other states enrolled under national mobility schemes.

REFERENCES

- [1] T. H. M. Kanta, A. V. V. Reddy, K. S. Reddy, N. Uma and R. Rahman, “AI-Powered Student Assistance Chatbot for Department of Technical Education Using NLP and ML,” *International Journal of Research in Applied Science & Engineering Technology*, vol. 13, no. 5, pp. 3232–3237, May 2025.
- [2] S. S. Bhosale, R. Mhatre, P. Patil and A. Shinde, “EduTech Portal: An AI-Powered Student Assistant Chatbot,” *International Journal of Innovative Science and Research Technology*, vol. 10, no. 5, pp. 122–133, May 2025.
- [3] S. Saifi, A. Hussain and M. Ansari, “AI-Powered Student Assistance Chatbot for Technical Institutions,” *International Journal of Computational Learning & Intelligence*, vol. 4, no. 4, pp. 823–831, 2025.
- [4] N. F. Davar, M. A. A. Dewan and X. Zhang, “AI Chatbots in Education: Challenges and Opportunities,” *Information*, vol. 16, no. 3, Article 235, 2025.
- [5] A. Gregorac, B. Br  nner and M. Ebner, “Chatbots in Education: A Systematic Rapid Literature Review,” in *Proc. SITE International Conference*, Orlando, FL, USA, Mar. 2025.
- [6] S. Wollny, J. Schneider, D. Di Mitri, J. Weidlich, M. Rittberger and H. Drachsler, “Are We There Yet? A Systematic Literature Review on Chatbots in Education,” *Frontiers in Artificial Intelligence*, vol. 4, Article 654924, 2021.
- [7] F. Clarizia, F. Colace, M. Lombardi, F. Pascale and D. Santaniello, “Chatbot: An Education Support System for Student,” in *Proc. Int’l Symposium on Cyberspace Safety and Security*, Amalfi, Italy, 2018, pp. 291–302.
- [8] W. Almansoori, M. Sartawi and A. Shaalan, “ChatGPT as an Academic Advisor,” in *Proc. ISDA International Conference*, 2023.
- [9] Omnimind.ai, “AI Chatbots Reduce Administrative Workload in Education,” *Omnimind Technical Blog*, 2025.
- [10] Anthropic, “Claude API Documentation,” 2025. [Online]. Available: <https://docs.anthropic.com>
- [11] Maharashtra State Board of Technical Education (MSBTE), Official Portal, 2025. [Online]. Available: <https://msbte.org.in>
- [12] Directorate of Technical Education Maharashtra, Official Portal, 2025. [Online]. Available: <https://dtemaharashtra.gov.in>