iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79548

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Student Problem Solving and College Info Chatbot

Yash Ghadge[1], Pranav Dalvi[2], Suyash Chavan[3], Prof. Aasifa Bagwan[4]
[1, 2, 3]*Department of Computer Engineering, KBP Polytechnic, Satara, Maharashtra, India*

Abstract: ***This paper presents the design and development of a College Information Chatbot System, an intelligent web-based application built to automate the handling of student queries, complaints, and feedback at KBP Polytechnic, Satara. The proposed system employs rule-based Natural Language Processing (NLP) techniques to interpret user input and deliver real-time, accurate responses related to admissions, fee structures, departments, courses, and campus facilities. The system integrates a structured complaint management module with ticket generation, an OTP-based authentication mechanism, a feedback collection interface, and an admin dashboard for complaint tracking and resolution. Additionally, the chatbot is extended to WhatsApp via the Twilio API, significantly improving accessibility. The backend is developed using Python Flask, with Firebase Realtime Database for cloud-based data storage. Evaluation through 30 functional test cases confirms that the system handles valid and invalid inputs correctly, maintains stable performance, and satisfies the primary objectives of automation, security, and usability. The system substantially reduces manual workload on administrative staff while improving student experience and institutional communication.***

Keywords: Chatbot, Natural Language Processing, Complaint Management, College Information System, Flask, Firebase, WhatsApp API, OTP Authentication, Student Query Handling.

I. INTRODUCTION

The rapid advancement of information technology has transformed communication across all sectors, including education. Students today expect instantaneous access to accurate information regarding institutional services such as admissions, fee structures, available departments, academic schedules, and campus facilities. However, most educational institutions continue to rely on manual query-handling processes, which are inherently slow, error-prone, and resource-intensive.

A College Information Chatbot System addresses these limitations by providing an automated, intelligent virtual assistant that interacts with students in real time. The system utilizes Natural Language Processing (NLP) techniques to interpret user queries and generate relevant responses, eliminating the need for students to navigate multiple web pages or wait for administrative assistance.

Beyond query resolution, the proposed system incorporates a structured complaint management module that allows students to report issues, receive unique ticket IDs, and track resolution status. Feedback collection is also integrated, enabling continuous improvement of institutional services. The system is accessible through a web interface and is extended to WhatsApp via the Twilio API, ensuring broad accessibility.

This paper presents the complete design, methodology, implementation, and evaluation of the College Information Chatbot System developed at KBP Polytechnic, Satara. The sections that follow cover the problem background, objectives, literature review, system architecture, design methodology, results, and conclusions.

II. PROBLEM STATEMENT

The existing approach to student query management at most polytechnic institutions is predominantly manual. Students are required to physically visit administrative offices or sift through static web pages to find information related to admissions, fees, or courses. This process is time-consuming for students and resource-intensive for staff.

Complaint submission follows a similarly inefficient path: students file issues via paper forms or emails, with no reliable mechanism for tracking resolution or ensuring accountability. The absence of a centralized, automated communication platform results in inconsistent responses, delayed resolutions, and growing dissatisfaction.

The core problem can be summarized as follows:

1) No real-time query response mechanism exists for students.
2) Complaint management lacks structure, tracking, and transparency.
3) Administrative staff face excessive manual workload.
4) No integration exists with modern platforms such as WhatsApp.
5) Feedback collection is unsystematic and rarely actionable.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

III. OBJECTIVES

The College Information Chatbot System is designed with the following primary objectives:

1) To develop an automated chatbot capable of handling diverse student queries related to admissions, fees, courses, departments, and campus facilities.
2) To implement a structured complaint submission module with unique ticket ID generation and status tracking.
3) To ensure data security through OTP-based authentication for sensitive operations.
4) To collect and store user feedback for ongoing system improvement.
5) To develop an admin dashboard for centralized complaint management and resolution.
6) To integrate the chatbot with WhatsApp via the Twilio API for extended accessibility.
7) To utilize Firebase Realtime Database for efficient, scalable data storage and retrieval.
8) To build a system that is scalable, reliable, and extensible for future enhancements.

IV. LITERATURE REVIEW

A. *Chatbot Technologies and Applications*

Vyas et al. (2018) conducted a comprehensive survey on chatbot systems and technologies, examining both rule-based and AI-based approaches across domains including education, business, and customer support [1]. Their work demonstrated that rule-based systems offer predictable, fast responses but lack flexibility for complex, ambiguous queries. AI-based systems, while more capable, require large annotated training datasets.

Reshmi and Balakrishnan (2019) proposed an intelligent chatbot using AI and NLP that could handle multiple simultaneous user interactions [2]. Their system highlighted the advantages of intent detection in improving response accuracy and reducing administrative overhead. However, the complexity of implementation and dependency on training data remained key challenges.

B. *FAQ and Educational Chatbots*

Kumar et al. (2020) developed an FAQ chatbot using NLP that matched user queries against a predefined dataset to provide instant automated responses [3].

While the system was easy to implement and demonstrated fast response times, it was constrained to predefined questions and lacked learning capability.

Sharma et al. (2021) designed a chatbot specifically for educational institutions to handle student queries related to courses, admissions, and fees using AI techniques [4]. The system provided 24/7 assistance and reduced staff workload, but its performance was heavily dependent on training data quality, and it exhibited limited capability in handling complex academic queries.

C. *Helpdesk and Complaint Management Systems*

Patel and Verma (2020) integrated chatbot technology with helpdesk services to provide automated support for query and complaint handling [5]. Their system allowed unresolved issues to be escalated for manual review. Although it reduced response time and improved issue management, its handling of complex complaints remained limited.

Singh and Mehta (2019) proposed a web-based complaint management system where users submit complaints through a web interface stored in a database for administrator review [6]. The system offered organized complaint tracking and reduced paperwork, but lacked real-time interaction and depended heavily on manual administrative action, resulting in slower resolutions.

D. *Research Gap*

The review of existing literature reveals several critical gaps: the absence of a unified system combining chatbot query handling, structured complaint management, and feedback collection; no real-time complaint tracking; limited platform integration; and lack of secure authentication mechanisms. The proposed system addresses all these gaps within a single, integrated platform.

V. SYSTEM ARCHITECTURE

A. *Architecture*

The system follows a three-tier architecture, which separates the system into independent layers for better scalability, maintainability, and performance.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

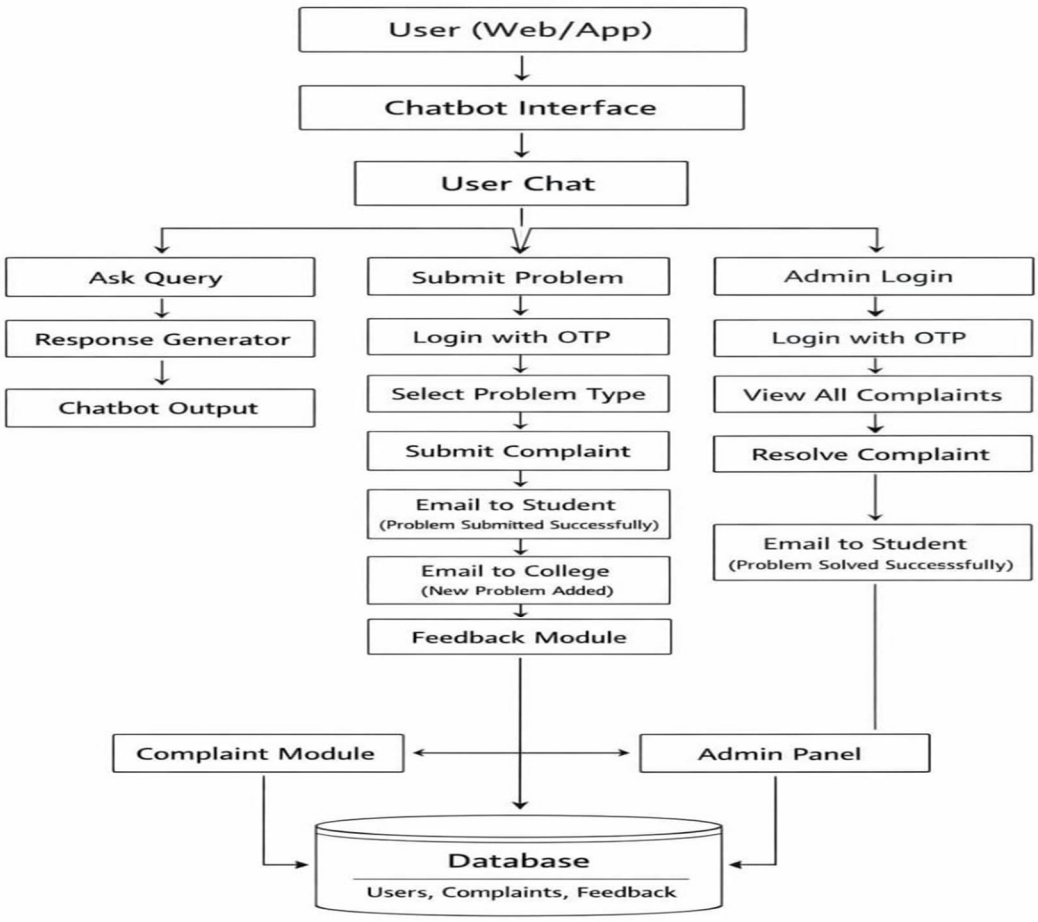


Fig v : System Architecture

B. User Interface Layer

This layer encompasses both the web-based chatbot interface and the WhatsApp interface powered by the Twilio API. It accepts user inputs such as queries, complaints, and feedback, displays chatbot responses, and provides navigational options. Users can interact without mandatory login for general queries, reducing friction and improving accessibility.

C. Application Layer

The core processing layer is built on a Python Flask backend API. It houses the Chatbot Engine responsible for intent detection, the Query Processing Module, the Complaint Management Module, and the Feedback Management Module. The layer receives requests from the UI, routes them to appropriate modules, validates and formats data, and generates dynamic responses.

Intent detection logic parses user input and matches it against predefined patterns to identify the appropriate response category: admission, fees, courses, location, complaint, feedback, or admin action.

D. Database Layer

Firebase Realtime Database serves as the cloud-based NoSQL backend. It stores complaint records (including user email, description, category, status, and timestamp), user feedback, and the predefined chatbot knowledge base. Real-time synchronization ensures that status updates by administrators are immediately visible to users.

E. System Flow

The system flow begins when a user accesses the chatbot interface. For general queries, the intent is detected, relevant data is retrieved from the knowledge base, and a response is generated instantly. For complaint submission, the user authenticates via OTP, provides complaint details, and receives a unique ticket ID. The admin reviews all complaints via the dashboard and updates their status upon resolution, triggering an email notification to the user.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. SOFTWARE DESIGN AND METHODOLOGY

A. Tools and Technologies

The system is implemented using the following technology stack: Python as the primary programming language for backend logic; Flask as the micro web framework for API development; HTML, CSS, and JavaScript for the frontend interface; Firebase Realtime Database for cloud storage; and the Twilio WhatsApp API for messaging integration. The development environment supports Windows and Linux platforms, and the application is compatible with modern web browsers including Google Chrome, Microsoft Edge, and Mozilla Firefox.

TABLE I
TOOLS AND TECHNOLOGIES USED

Sr.	Tool / Technology	Purpose
1	Python	Backend development and chatbot logic
2	Flask	Backend API and HTTP request handling
3	HTML/CSS/JS	Frontend interface and interactivity
4	Firebase Realtime DB	Complaint, feedback, and chatbot data storage
5	Twilio WhatsApp API	WhatsApp chatbot integration

B. Modules

The system comprises eight functional modules:

1) User Authentication Module: Manages user login and signup using OTP-based verification to ensure that only authorized users access sensitive features such as complaint submission.
2) Chatbot Module: Acts as the primary interface between the user and the system. It guides users, detects intent, and returns real-time responses for a wide range of queries.
3) Complaint Management Module: Enables users to submit detailed complaints with problem type, description, and expected resolution. Each complaint receives a unique ticket ID for tracking.
4) Admin Panel Module: Provides administrators secure access to view, analyze, and update all complaint records. It includes graphical status summaries and detailed complaint tables.
5) Email Notification Module: Sends automatic email alerts to users upon complaint submission and upon resolution, ensuring transparent and timely communication.
6) Feedback Module: Collects ratings and comments from users after interaction, enabling continuous system improvement.
7) OTP Verification Module: Generates and validates one-time passwords to secure login and complaint submission processes.
8) Database Module: Manages all data storage and retrieval operations via Firebase, ensuring consistency, real-time updates, and data integrity.

C. Algorithm and Flowchart

The system follows a structured algorithmic flow:
Step 1: Start
Step 2: Open the chatbot interface
Step 3: Display greeting message to the user
Step 4: User enters query or selects an option
Step 5: System checks the user input
Step 6: If input is valid:
    Fetch relevant information (Admission / Fees / Courses / Location)
Step 7: Else:
    Display fallback message for invalid input
Step 8: If user selects Feedback:
    Accept feedback input
    Store feedback in database

Step 9: If user selects Report Problem:
Accept complaint and email
Store complaint in database
Step 10: If Admin Login is selected:
Verify email and OTP
Grant access to admin dashboard
View and update complaints
Step 11: Display response to user
Step 12: Ask user whether to continue
If Yes, go back to Step 4
If No, proceed to next step
Step 13: End

D. Security Considerations

Security is enforced through OTP-based authentication for sensitive operations, ensuring that only verified users can submit complaints or access the admin dashboard. Firebase security rules protect data from unauthorized read or write operations. The system does not expose raw database credentials in the frontend, and all API communications are conducted over HTTPS.

VII. RESULTS AND EVALUATION

A. System Output

The College Information Chatbot System was successfully implemented and evaluated at KBP Polytechnic, Satara. The chatbot accurately responds to queries related to admission procedures, fee structures (categorized by student category), available courses, campus location, and principal information. Screenshots of the deployed system confirm functional correctness across all major use cases including web chatbot interaction, WhatsApp integration, complaint submission, feedback collection, and admin dashboard operations.

B. Testing

A total of 30 functional test cases were designed and executed to validate system behavior against both valid and invalid inputs. All test cases passed successfully, confirming the correctness and robustness of the system.

TABLE II
SELECTED TEST CASES AND RESULTS

TC ID	Input	Expected Output	Status
TC01	Hello	Greeting message displayed	Pass
TC02	Admission process	Admission details returned	Pass
TC03	Fees	Fee structure displayed	Pass
TC04	Submit Problem	Complaint stored in DB	Pass
TC07	Admin Login	Dashboard accessed	Pass
TC10	Invalid Input	Fallback message shown	Pass
TC18	OTP Correct	Login granted	Pass
TC19	OTP Wrong	Access denied	Pass
TC30	Mobile View	Responsive UI rendered	Pass

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. Performance

The system demonstrated stable and responsive performance under all tested conditions. Response generation was near-instantaneous for chatbot queries, and the admin dashboard loaded complaint data without noticeable delay. The Firebase Realtime Database ensured that status updates by administrators were reflected immediately on the user side. No critical bugs or crashes were observed during testing.

VIII. CONCLUSION

This paper presented the design, development, and evaluation of a College Information Chatbot System that automates student query handling, complaint management, and feedback collection at KBP Polytechnic, Satara. The system successfully replaces inefficient manual processes with an intelligent, accessible, and secure digital platform.

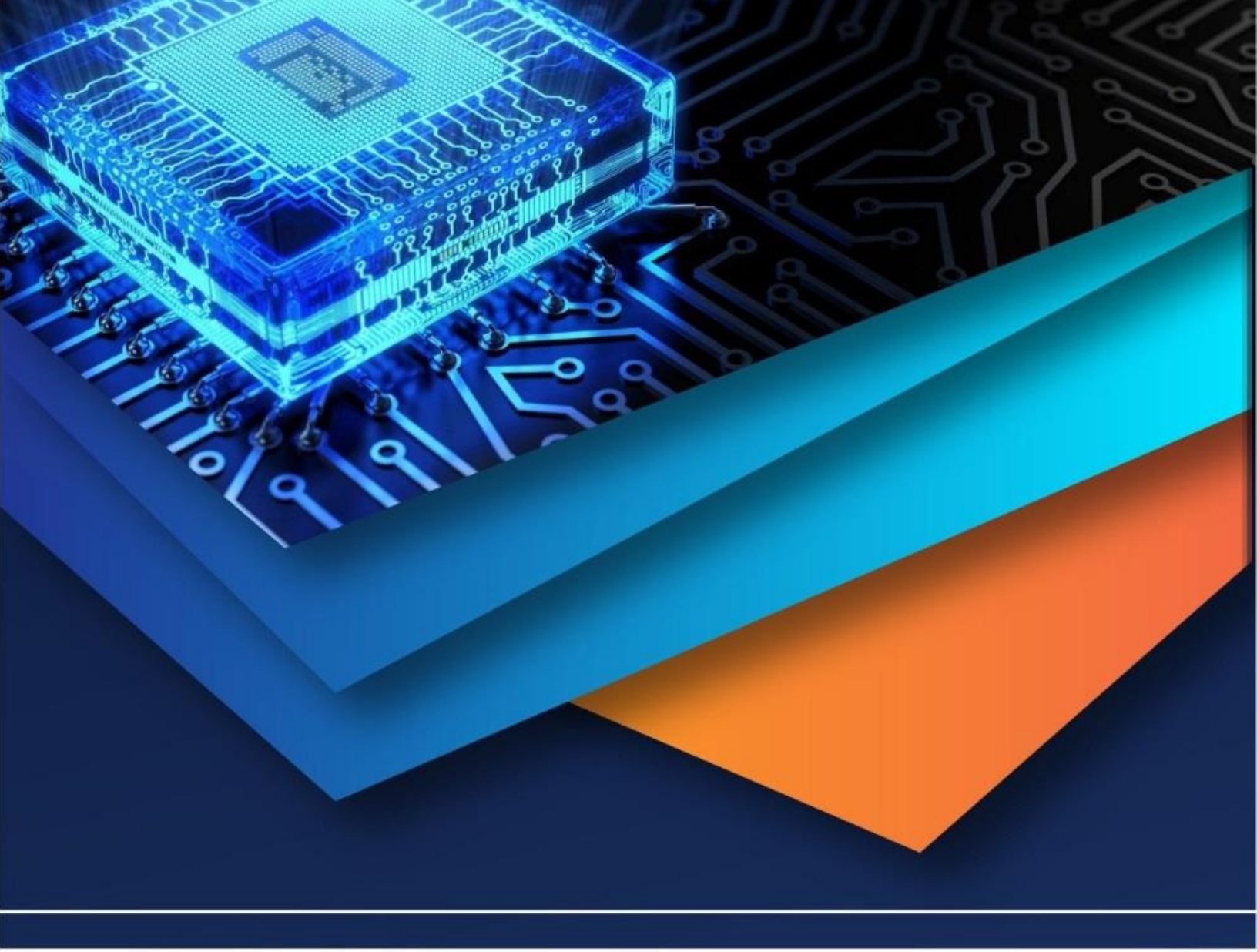
The chatbot delivers instant, accurate responses to a wide range of student queries without requiring login, thereby reducing barriers to access. The structured complaint module with OTP authentication, ticket generation, and admin-driven resolution ensures transparent and accountable issue handling. Email notifications keep users informed throughout the process, and the admin dashboard provides administrators with comprehensive control and visibility.

All 30 functional test cases passed, validating system correctness, stability, and usability. The WhatsApp integration via Twilio further extends the system's reach to mobile users on a familiar platform.

Future enhancements include integration of machine learning and advanced NLP for improved intent recognition, development of a dedicated mobile application, full WhatsApp Business API deployment, voice-based interaction support, and integration with institutional ERP systems for real-time data synchronization.

REFERENCES

[1] K. Vyas, N. Sharma et al., "A Survey on Chatbot Systems and Technologies," IEEE, 2018.
[2] S. Reshmi, K. Balakrishnan et al., "Design and Implementation of Intelligent Chatbot System," IEEE, 2019.
[3] R. Kumar, P. Singh et al., "Development of FAQ Chatbot Using Natural Language Processing," IEEE, 2020.
[4] A. Sharma, V. Gupta et al., "Chatbot for Student Support System Using AI," IEEE, 2021.
[5] M. Patel, S. Verma et al., "Smart Helpdesk System Using Chatbot Technology," IEEE, 2020.
[6] N. Singh, R. Mehta et al., "Web-Based Complaint Management System," IEEE, 2019.
[7] Twilio Documentation, WhatsApp Messaging API. Available: https://www.twilio.com/docs
[8] Google Firebase Documentation. Available: https://firebase.google.com/docs
[9] Flask Documentation, Pallets Projects. Available: https://flask.palletsprojects.com/
[10] Python Software Foundation, Python 3 Documentation. Available: https://docs.python.org/3/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)