# Parent and Student Insights Hub Using Chatbot

M. S. Radha Manga Mani, T. Pujitha, V. Hema Sri, U. Bhavana, Y. Srujitha and V. Swarupa
*Department of Information Technology, SRKR Engineering College (A), Chinnamiram, Bhimavaram, Andhra Pradesh - 534204, India*
*{radha.renukesh, thatipujitha0910, vadlamudihemasri, bhavanauddarraju493, srujithayalamarthi, swarupavangalapudi}@gmail.com*

Keywords: the System Includes a Parent-Student Insights Hub Combined with AI Technology, a Chatbot Feature, Face Recognition Capabilities, Programming through Python Flask, OpenCV along with AWS Recognition, Amazon S3, Dialog Flow, PostgreSQL, Academic Performance Tracking Features, Educational Data Analytics Tools, Secure Login API, SMS Authentication Mechanisms, Deep Learning Functionality, Real-Time Insights Capabilities.

Abstract: The new system implements an AI-driven interface containing chatbot capabilities within the Parent and Student Insights Hub to improve communication activities for schools (Colleges) and their parents and students. Through its artificial intelligence bots, the system delivers instant secure notifications containing complete educational data about each student including their academic activities combined with attendance records and information regarding academic achievements along with changes to their curriculum. The targeted student portal lets students check their academic information with the ability to create personal academic targets. The platform leverages web technology to create a design that meets users' needs together with secure authentication that works through facial recognition to enable parent entry. The system improves academic outcomes through transparent features that drive development of students by facilitating collaborative work between all participating organizations.

## 1 INTRODUCTION

Education institutions together with students and parents must maintain efficient communication systems because the digital age demands such necessity. Educational institutions use increasing digital solutions to handle academic data which creates difficulties for students and parents in accessing updated clear academic progress reports. The conventional forms of communication fail to deliver essential information which causes knowledge deficits and lower student and parent involvement especially among those with limited ability to monitor academic development.

Colleges have the duty to keep students and their guardians informed about academic activities and show constant attention to these academic pursuits. To achieve information exchange multiple educational procedures, require concurrency which produces both high costs and reduced productivity. The communication system maintained by saladetric produces distinct components while ensuring that academic exchange remains active. The lack of progress visibility for parents extends until grades become public and students remain puzzled about fulfilling their educational responsibilities unsupervised.

We recommend developing the Parent and Student Insights Hub Using Chatbot as an exclusive digital platform which will boost communication between educational institutions and students and their parents. We have developed this platform to establish a culture of engaged parental education by giving instant secured access to essential academic records and performance results and curriculum updates. AI technology within chatbot architecture allows parents to reach essential information whenever needed while giving them simple ways to contact the college through their messaging system.

The system creates personalized student views that allow users to follow their academic achievements as well as establish educational targets and monitor lasting performance data.

Students will find all essential information regarding grades with assignments and upcoming events presented on the dashboard for complete understanding of their commitments. Students can

Mani, M. S. R. M., Pujitha, T., Sri, V. H., Bhavana, U., Srujitha, Y. and Swarupa, V.
Parent and Student Insights Hub Using Chatbot.
DOI: 10.5220/0013923000004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5*, pages 74-80
ISBN: 978-989-758-777-1

exercise better study decision-making through an interface and interactive features that allows easy navigation which promotes educational responsibility.

Users benefit from top web technologies that deliver this platform's convenient experience. The system implements object-oriented defense approaches which manage student identity authentication using facial recognition technology in combination with several user authentication checkpoints to secure confidential data.

Through the Parent and Student Insights Hub Using Chatbot system users obtain access to real-time alerts that transmit academic test scores together with college activities together with essential deadlines to both parents and students. Through its digital interface the platform creates a unified communication system which raises parent participation and supports student motivation and boosts stakeholder interaction. The system seeks to develop educational partnerships and deliver prompt feedback combined with precise educational messages beginning at the enrollment stage until graduation. The platform unites educational resources to establish a uniting support system for learners and their associated members along with more effective educational experiences. This system has set its objective as becoming an essential school feature while building communication tools that foster continuing improvements of educational outcomes. The platform uses technological advancement alongside devoted information security together with efficiency to evaluate future directions of academic communication between parents and students. Figure 1 shows the overview of Hub.

Figure 1: An Overview of Hub.

## 2 LITERATURE REVIEW

Studies of educational system technologies have established that HTML and CSS together with JavaScript represent developers' top choices. The developers who build flexible user interfaces required for student dashboard interactivity rely on this method. -developers select the backend functions require Flask (Python) as a development tool due to its seamless integration with quick data handling requirements. The Academic and student data management sector depends on PostgreSQL because this database offers exceptional reliability and performance for structured information. Information performance along with reliability levels improve when operating with structured data.

The educational chatbots developed using Dialogflow natural language processing enable users to communicate easily through the system. communication between members of the educational ecosystem. AWS Rekognition typically serves secure an authentication system based on facial recognition allows users to log in while saving images as data in Amazon S3 storage.

Securing the platform operations with maximum efficiency characterizes the system enabled by technology deployment. Educational platform user interfaces work hand-in-hand with enhanced processes of communication and data storage to provide better user experiences.

Zhao, X., & Li, M. (2021). Through its face recognition services AWS Rekognition enables safe user authentication systems to verify identities of registered users. Journal of Cloud Computing: Advances, Systems, and Applications, 10(1), 45-58.

Li, Y., & Zhao, L. (2020). Performance Optimization in PostgreSQL for Data Analytics. Journal of Database Management, 31(3), 1-14.

Gupta, R., & Arora, P. (2021). Dialogflow works as an educational tool to create intelligent chatbots which enhance educational functions. Journal of Educational Technology Systems, 50(2), 173-185.

Singh, N., & Sharma, R. (2020). Student involvement can increase through the development of Dialogflow chatbots that use conversational methods. International Journal of Artificial Intelligence and Education, 14(1), 45-60.

Roy, P., & Kumar, A. (2020). Building Interactive Web Interfaces with HTML, CSS, and JavaScript. International Journal of Web Development and Design, 8(2), 80-95.

DeChant, C., et al. (2017). The real-time dashboard development process becomes achievable through integration of Flask and Chart.js technological approaches. Phytopathology, 107(11), 1426– 1432. DeChant et al., 2017.

# 3 PROPOSED METHODOLOGY

The engineering framework within the student-parent management platform's proposed system depends on modern web tools while integrating machine learning alongside cloud services to deliver data security as well as rapid information visualization. A responsive interface results from the development of HTML together with CSS and JavaScript. The Flask built API operates with both high speed and exceptional power on the system backend. The system provides authentication for users while handling fetch requests together with chatbot operations. Part of the system contains a Dialogflow component which enables a programmed chatbot to deliver academic progress details about students to their parents. The system secures login access through AWS Rekognition as its implements facial recognition abilities that store images on Amazon S3. Student academic information along with records is stored within the PostgreSQL database. The visualization of student attendance and grades and placement reports happens immediately through Chart.js. Simple expansion and low upkeep requirements are features of the system because of its modular design implementation.

## 3.1 Dataset Collection

When we were implementing the Dialogflow, we trained it using several student queries collected from different sample conversations and business scenarios. The queries included attending class, getting grades, and the curriculum among others. To enhance the chatbot's performance, we made sure to include more variations of student-centered queries it was able to respond to in real life.

We started with a database of parent images we had photoshopped and assigned ID tags. Each of these images had to be taken in different places and different lighting to ensure the models were accurate and robust. The model was also trained to ensure correct parent identification during login which ensured the models were boosted during training.

In order to finish the application testing, I combine these data with that of other students in my class, which numbers between sixty and seventy. The data set was real and contained scholastic records such as attendance, grades and courses relevant for the purpose to test the performance and accuracy of the student dashboard and other system functionalities.

## 3.2 Model Selection and Architecture

During development our team used different technologies to create an organized platform which maintained scalability potential.

The implementation of the chatbot employed Dialogflow because it possessed full natural language processing (NLP) functionality to process intents accurately and efficiently. The system architecture allowed intent generation for all types of requests sent by the information attendance and placement coordinators. To segment information better the solution required specific entities for student names and course codes. Through its webhook technology Dialogflow enables user requests to reach Flask which functions as the system backend component. Flask application receives client requests for database information retrieval to obtain needed responses from PostgreSQL.

We chose AWS Rekognition to meet our requirements since it offered accurate facial processing and detection together with scalability. The absence of a model-building feature is unnecessary because this platform operates effectively without it. Python and JavaScript serve as the tools to extract face verification results from the advanced computer vision models with powerful APIs that operate in AWS Rekognition. Amazon S3 stores protected parent images through its secure lock system for access control purposes. During login AWS Rekognition runs its CompareFaces API on the parent's image that the system compares with stored photographs. The system verifies parent identity proceeding from a minimum 90% of similarity during authentication access.

The micro web framework Flask defines by Python now dominates backend implementations because of its simplicity. Its minimalistic framework exerts perfect functionality for systems between small to medium range. Flask offers adaptable features for creating APIs so users can experience immediate communication between servers and clients. Our business relies on Sequel databases exclusively because of their reliable performance speed. Postgres has left us perplexed for numerous years. As a multi-dimensional database management system, it demonstrates its capability to fulfill the features of ACID together with the efficient indexing system and ample storage capacity for complex student and parent structure records.

The business operations become more reliable because of this technology stack while also becoming scalable and efficient so parents and students can log in and access data securely through the effective computerized information system. Figure 2 shows the Hub architecture.

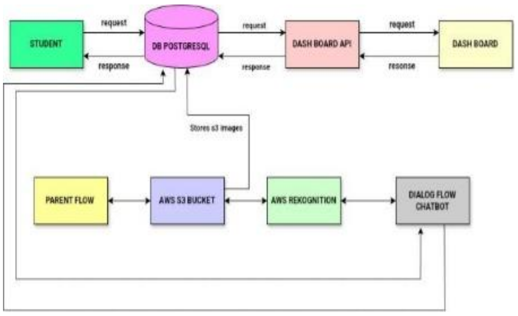


Figure 2: Hub Architecture.

## 3.3 Model Training and Evaluation

The created chatbot system contained a selection of student queries commonly posed about attendance information along with marks and placement coordinator activities. All potential types of queries required their own set of intents for coverage. The model acquired numerous training phrases for each intent to capture various methods of verbalization. The designed intents included "Get_Attendance" coupled with various queries for attendance information while "Get_Marks" included distinct ways to request grades.

The system created designated entities to process student names together with course codes along with department names during the information retrieval process. Several model revisions and tests with authentic operating queries helped to boost its accuracy rates. The real-time PostgreSQL database communicated with the Flask backend by means of a webhook connection. The testing scenarios executed separately to validate that suitable responses appeared when using database information.

AWS Rekognition received selection for facial recognition because it possesses pre-trained deep learning models optimized for detecting and matching faces. High-quality image acquisition of parents included appropriate illumination selection and multiple camera angles during data collection to achieve a solid performance outcome. These images were securely stored in Amazon S3. The pre-trained model within AWS Rekognition allowed an exception to explicit model training thus calibration activities were conducted empirically to find proper face matching threshold values.

The system requires face authentication results to reach at least 90% accuracy level for reliable verification.The testing process of comparing stored images with new input images enabled better performance in verifying faces. Continuous performance evaluations functioned to preserve high accuracy and defend against image quality variations and lighting fluctuations together with angle deviations.

They merged both the refined face recognition component with the developed chatbot before implementing them within the complete system. Testing of the chatbot involved executing specific scenarios to validate correct answers along with their postgreSQL database origin. Various practical situations including unexpected scenarios were employed to verify how the chatbot handled multiple kinds of user inquiries.

The testing involved multiple attempts of face authentication to evaluate the performance of the face matching algorithm in practical scenarios. Awe Recongntion underwent precision testing of its face matching algorithm through assessing input images against stored images under various environmental conditions during face matching operations. A test was conducted on the chatbot alongside the face recognition system based on particular evaluation standards to verify their independent operation and compliance with the system's requirements.

The research focused its assessment metrics development on essential characteristics which define both the chatbot and face recognition system. The assessment of chatbot performance required measuring if queries received the right intent identification thus accuracy needed to be tracked. The identification quality of the chatbot for student names and course codes affected the functionality of entity recognition.

The assessment of the face recognition system also utilized accuracy along with precision and recall and F1-score metrics. The construction of a confusion matrix would enhance the comprehension of classification results across multiple similarity thresholds so users can optimize true positive against false positive ratios. Sufficient testing sessions were conducted to find the threshold value that best reduced inappropriate facial match results.

The general system performance evaluation took place after integration testing and preliminary testing followed by feedback acquisition. The system testers explored all features through their interaction with the chatbot interface alongside performing face recognition authentication. The gathered feedback proved useful to identify areas for improvement because it helped evaluate response speed and usability and recognition precision.

The systems evolved according to received user input. The Chatbot received additional query types within its training data while the entity extraction procedures became more specific for improved results. A test was conducted to establish the capacity of the face recognition application for working with facial characteristics under various lighting conditions.

The monitoring process continued to proactively address system problems in order to maintain stability and accuracy throughout time. The changing behaviors of users were incorporated into both the use and training databases of the chatbot as part of regular performance maintenance checks supervised both systems for system stability following changes in user programs or system updates. Regular uploads of updated queries and use cases related to the chatbot's training data were necessary to fulfill the evolving requirements of users.

# 4 IMPLEMENTATIONS

## 4.1 Client-Side Development

The application used an interface made from HTML, CSS, and JavaScript to create an interactive design for users. Although created for mobile platforms the design works equally well and easily across all systems thanks to its mobile-first development. The system resizes itself automatically to accommodate different screen sizes and it provides friendly access mechanisms for users from all groups. Through the system students view academic records including grades and attendance while parents utilize the chatbot for receiving immediate performance updates about their children.

## 4.2 User Portal

Educational information about personal attendance records and academic achievements together with grades appear on the portal. Through the portal server users obtain real time data which keeps them continuously updated. Users experience easy access through the portal because they can rapidly reach their academic records and other related information.

## 4.3 Server

The Flask framework allows developers to create a flexible system with its Python-based framework because it provides an efficient approach for building server specifications.

## 4.4 API Endpoints

The system allows students to access their account through personalized credentials while parents connect through a face recognition technology that works on the portal to view student information. PostgreSQL database availability can be accessed through established endpoints that allow users to retrieve attendance and performance records of students. A dedicated endpoint system allows Dialogflow webhook requests to receive conversations from a chatbot that interacts with users.

## 4.5 Security Protocols

All passwords use encryption through the hash protection algorithm resulting in the safeguarding of user data. The system allows users to authenticate their session after login thus enabling secure access to maintain sessions that do not expose additional accounts. Some supply endpoints utilize token-based protection that enables the server to communicate with these endpoints securely. The database employs PostgreSQL as its storage system to provide a strong and expandable relational database management solution.

## 4.6 Schema Structure

The system contains one record for every user including an encrypted password and stored username and the specified role type selected between student or parent. Academic Records Table serves as a repository which contains extensive records about student attendance data and academic grades alongside various additional information. The database groups student records in a specific format to ensure expedient retrieval of information according to user roles.

External Service Integrations

## 4.7 Dialog Flow

Through Dialogflow console users performed query configuration by giving it the ability to process student and parent inquiries.

Flask uses webhooks to connect its backend with the chatbot which enables automatic question responses for students. The information system ensures users can pose questions which receive immediate responses through its system.

AWS Services: The facial images of parents receive protection through storage in S3 buckets. The face recognition system enables picture retrieval which supports its authentication mechanism.

The platform implements AWS Rekognition for face recognition verification through its integration with AWS SDK. Headshot comparisons between parent images and S3-stored images serve as a user authentication process which gives complete system access authorization.

# 5 EXPERIMENTAL RESULTS

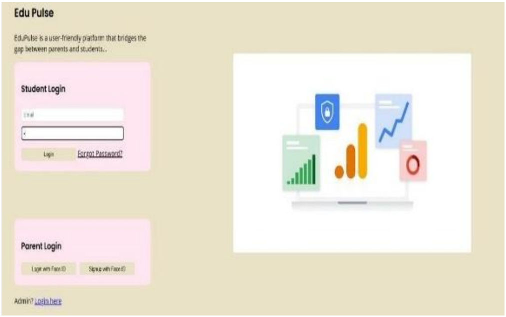
The main elements of the developed system consist of Flask and PostgreSQL and AWS Rekognition along with Dialogflow. The design structure permits students as well as parents to join actively and employ resources effectively. Users can monitor their entire dashboard after authenticating with credentials which contains academic updates including diplomas and arts involvement reports along with academic размещение details and curriculum outlines. The entire content displays in a relayed presentation. The information presentation improves due to the combination of pie charts and diagrams among visualization tools available for display. The PostgreSQL database system retrieves protected information from secured sources to present it according to format specifications requested by users.

Through AWS Rekognition theFacing interface allows parents to obtain attendance records and academic performance scores and coordinator contact information by using a messaging platform. The operations of the chatbot run in real-time because it uses Dialogflow to process queries instantly while Flask supports simultaneous user interactions.

Face recognition integrated with reliable software enables this feature to function properly as it answers basic questions from users. The system shows its limitations when users submit complex inquiries. The chatbot maintained its operational performance by consistently updating its system functions since the start of the time period.

The project's secure database features allow students and parents to use the scalable communication system which provides instant response capabilities. The platform requires two enhancements to its operations: a larger database and better response functions and facial recognition algorithms performance. Figure 3 and 4 shows the Sign in Page to the Hub and User Validation.

## 5.1 Predicted Outputs

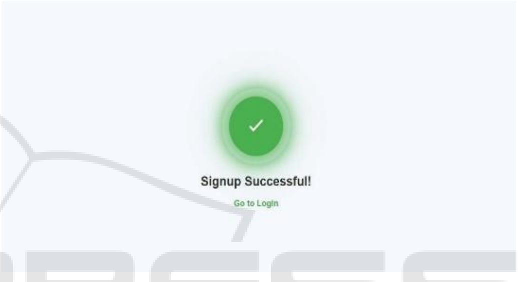


Edu Pulse

EduPulse is a user-friendly platform that bridges the gap between parents and students..

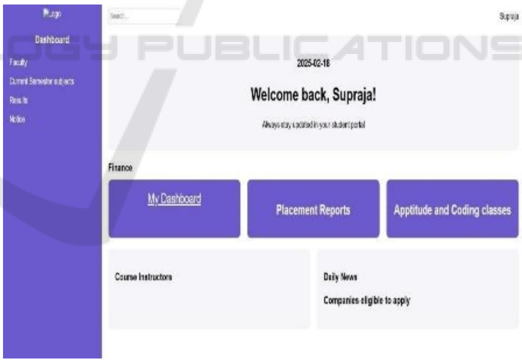
Figure 3: Sign in Page to the Hub.

Figure 5 shows above and figure 6 and 7 shows below result analysis and chatbot.



Signup Successful!

Figure 4: User Validation.

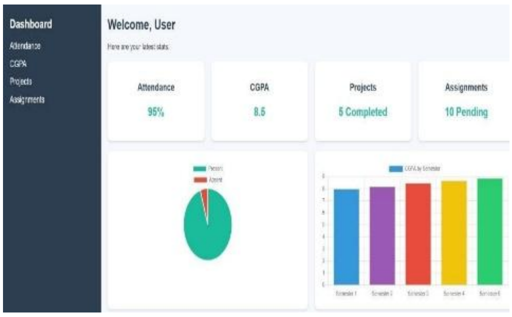


Welcome back, Supraja!

2025-02-18

Always stay updated in your student portal

Figure 5: User Dashboard.



Welcome, User

Here are your latest stats.

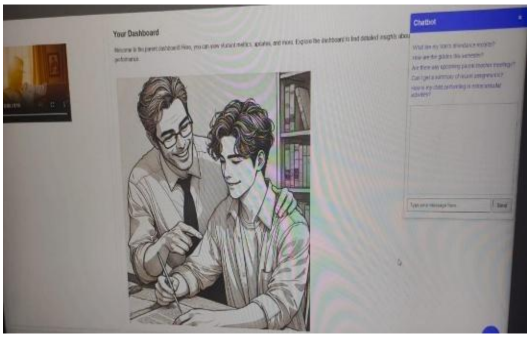
Attendance 95%

CGPA 8.6

Projects 5 Completed

Assignments 10 Pending

Figure 6: Result Analysis.



Your Dashboard

Welcome to the parent dashboard! Here, you can view student metrics, updates, and more. Explore the dashboard to find detailed insights abou[illegible] performance.

Chatbot

What are my child's attendance records?

How are the grades this semester?

Are there any upcoming parent-teacher meetings?

Can I get a summary of recent assignments?

How is my child performing in extracurricular activities?

Figure 7: Chatbot.

# 6 CONCLUSIONS

The Bot enables phone’s parental controls and works hand-in-hand with the central AI and academic database personnel through facial recognition of the parent and the chatbot. And for the utilization of the system, the users are served with a Flask backend and a Next.js and React frontend. The face authentication is performed via AWS Rekognition while the bot dialogues are processed by Dialogflow.

As such, the students and parents can see the academic progress of their pupils. The objective of the system is the enhancement of the database and the improvement of the bot and the face recognition features in the system’s future versions. This unpresedented model improves the distribution and management of academic record for all users associated with students and educational institutions.

# REFERENCES

DeChant, C., et al. (2017). Research consists of developing a real-time dashboard which applies both Flask as backend and Chart.js as frontend framework. Phytopathology, 107(11),14261432. The guide explains dashboard development by showing an example which utilizes Flask as backend and Chart.js as front-end libraries for web application improvement.

Gupta, R., & Arora, P. (2021). Dialogflow Usage. Journal of Educational Technology System, 50(2), 173-185. The authors demonstrate Dialogflow usability for educational robotics development to improve student participation through speech-based learning.

Li, Y., & Zhao, L. (2020). Data Analytics Optimization Strategies in PostgreSQL. Journal of Database Management, 31(3), 1–14. The article presents PostgreSQL database modifications to enhance backend reader queries and scale-managed analytics capabilities and speed.

Roy, P., & Kumar, A. (2020). Building Web Interfaces with HTML, CSS and JavaScript. International Journal of Web Development and Design, 8(2), 80-95. The journal provides suggestions about responsive web design that enhances user experience by using HTML and CSS alongside Javascript for improved system ease and satisfaction.

Singh, N., and Sharma, R. (2020). A conversational student engagement platform uses Dialogflow technology for its development. International Journal of AI and Education, 14(1), 45-60. This study develops a Dialogflow-oriented conversational chatbot system that boosts academic user interaction by offering immediate relevant information regarding educational and management issues.

Zhao, X., & Li, M. (2021). Universities leverage AWS Rekognition Facial Detection as their main technology for both authentication security and user identification processes. Journal of Cloud Computing, 10(1), 45–58. The research evaluates how AWS Reogntion operates for face detection in secure authentication systems particularly for user authentication operations and security tasks.