

Integrated College Web Portal with Event Management and Chatbot

V.S.Kavii¹, Dr. A. Bamini²

¹*MCA, Department of computer Applications, The Standard Fireworks Rajaratnam College for Women
Sivakasi, Tamil Nadu, India*

²*Head, Department of computer Applications, The Standard Fireworks Rajaratnam College for Women
Sivakasi, Tamil Nadu, India*

Abstract—The widespread adoption of digital infrastructure in academic institutions has created a pressing need for consolidated, intelligent web platforms. This paper presents the design and implementation of an Integrated College Web Portal with Event Management and Chatbot Support, developed to streamline administrative operations and enhance student engagement. The system employs HTML, CSS, and JavaScript for the front-end layer, Python Flask as API, and Google Sheets as a cloud-native data store. A conversational agent built on Google Dialogflow provides natural-language query resolution around the clock. The portal centralizes institutional content management, automates event registration and confirmation workflows, and delivers role-based access control for administrators and students. Agile development practices guided the project, and rigorous testing confirmed satisfactory performance, accuracy, and usability. The modular architecture supports seamless scalability for larger institutions and future feature additions.

Index Terms—College web portal, event management, chatbot, Dialogflow, Python Flask, Google Sheets, role-based authentication, cloud computing.

I. INTRODUCTION

The growing reliance on digital tools in higher education has made it imperative for institutions to adopt integrated platforms that consolidate communication, administration, and student support. Traditional processes—paper-based notice boards, fragmented email chains, and disconnected spreadsheets—create fertile conditions for miscommunication, delayed information, and elevated administrative workloads.

This paper presents the Integrated College Web Portal with Event Management and Chatbot Support, a comprehensive solution designed for The Standard Fireworks Rajaratnam College for Women (SFRC), Sivakasi. The platform consolidates management of website content, student records, institutional announcements, faculty profiles, and event coordination into a single, unified interface.

A core feature of the platform is an intelligent conversational agent built on Google Dialogflow. This chatbot enables students and visitors to retrieve information instantly without navigating multiple pages or contacting administrative staff. The assistant operates continuously, responding to queries about admissions, events, schedules, and general college information, thereby reducing operational burden on office personnel.

A. Google Dialogflow

Dialogflow is a natural language processing (NLP) platform developed by Google that designs and builds conversational interfaces such as chatbots and voice assistants. It interprets user input through intent-based processing, extracts entities from queries, and maintains conversation context for accurate multi-turn responses—capabilities that make it well suited for the student-facing support role in this project.

II. LITERATURE SURVEY

The development of integrated digital platforms for academic institutions has attracted growing research interest over the past decade. Prior work can be grouped into four thematic areas directly relevant to the proposed system: web-based college information

portals, automated event and registration management, AI-powered chatbot assistants in education, and cloud-based lightweight backend architectures.

A. Web-Based College Information Portals

Early college portal systems focused primarily on static content delivery, providing department listings, faculty directories, and notice boards with no real-time update capability. Subsequent work introduced content management interfaces that allowed non-technical administrators to publish updates directly, eliminating dependence on web developers for routine changes. Researchers noted that centralising institutional information in a single authoritative source significantly reduced the problem of data inconsistency across channels such as notice boards, printed materials, and email newsletters. The present system builds on this foundation by coupling content management with live event data and an automated chatbot, ensuring that every information channel remains synchronised without additional staff effort.

B. Online Event Registration and Management Systems

Research on academic event management systems consistently highlights manual paper-based registration as the primary source of errors, missed deadlines, and organisational overhead. Studies comparing manual and automated registration workflows found that digitised systems substantially reduce data entry errors and enable organisers to monitor participation in real time. Cloud-based spreadsheet tools such as Google Sheets have been adopted in several lightweight institutional systems as an alternative to heavyweight relational databases, offering instant accessibility, collaborative editing, and straightforward API integration without infrastructure costs. The event module of the proposed portal leverages these findings by implementing an online registration form backed by Google Sheets, with Flask enforcing capacity limits and triggering automated confirmation emails upon each successful submission.

C. Conversational Agents in Educational Portals

The integration of AI-powered chatbots into educational web portals has been an active area of development in recent years. Implementations based on rule-based engines were superseded by natural

language processing platforms that interpret free-form student queries and map them to predefined intents. Google Dialogflow has emerged as a widely adopted platform for this purpose because of its robust intent recognition, entity extraction, context handling, and seamless embedding into web pages via a JavaScript snippet. Studies deploying Dialogflow in university portals reported that a well-trained agent could resolve the majority of routine student enquiries covering admissions, timetables, and campus facilities without human intervention, freeing administrative staff for higher-value tasks. The chatbot component of the proposed system draws directly on these findings, with intents trained on representative SFRC-specific queries.

III. SYSTEM ARCHITECTURE

The proposed portal is designed on a client-server architecture that integrates frontend technologies, a backend server, and a cloud-based database to deliver an efficient, responsive web application.

A. Hardware and Software Environment

The system was developed and tested on an Intel Core i5 processor (2.50 GHz) workstation running Windows 10 Pro with 4 GB RAM and a 250 GB hard drive. Browser compatibility was verified on the latest versions of Google Chrome and Mozilla Firefox. An active internet connection is required for Google Apps Script and Dialogflow integration.

The software stack comprises HTML, CSS, and JavaScript for the frontend; Python Flask for the backend; Google Sheets as the data store; Visual Studio Code as the development environment; and Dialogflow (Google) as the chatbot platform.

B. Frontend Technologies

The frontend is built with HTML for structural markup, CSS for responsive visual presentation across desktops, tablets, and smartphones, and JavaScript for client-side interactivity including form validation and dynamic content updates without page reload.

C. Backend Technologies

Python Flask serves as the lightweight, micro-framework backend. Its routing system maps URLs to Python functions, its Jinja2 template engine renders dynamic HTML pages, and its RESTful API support

facilitates communication between the frontend and backend. Flask integrates readily with Google Sheets via the Google Apps Script API, enabling cloud-native data persistence without dedicated server infrastructure.

D. Working Principle

When a student visits the portal, the HTML/CSS/JavaScript frontend renders the interface and dispatches JavaScript fetch requests to the Flask backend. Flask authenticates the user, processes the request (event listing retrieval, registration submission, or query forwarding), communicates with Google Sheets to read or write data, and returns a JSON response. The frontend then renders the result—displaying event details, confirming registration, or displaying the chatbot reply. The Dialogflow agent operates in parallel, receiving natural-language input from the embedded chat widget and returning intent-matched responses independently of the Flask pipeline.

IV. METHODOLOGY

1) Requirements Analysis

Stakeholder interviews with SFRC administrative staff and students identified three principal problem areas: scattered institutional information with no central access point; manual, error-prone event registration processes; and the absence of automated query resolution, forcing students to contact staff for routine enquiries.

2) System Design

The Data Flow Diagram maps information flows from the student-facing frontend through the Flask backend to Google Sheets and back. The Use Case Diagram identifies two actor roles—Admin and Student/User—interacting with modules including Login/Authentication, Event Management, View Registrations, Chatbot Query, and Receive Confirmation Notification. The Class Diagram defines five primary classes: User, Frontend (UI), Flask Backend, Google Sheets DB, Event, Competition, and

Registration, with associations that mirror the runtime data flows.

3) Database Design

Three Google Sheets tables underpin the system. The Events table stores event_id, department, title, date, description, image filename, and fees. The Competitions table links competition names and capacities to their parent event via event_id. The Registrations table captures a timestamp, event_id, event title, student name, email, roll number, registration number, class, department, and selected competition for each enrolment.

4) Module Development

Development followed agile sprints. Admin modules—Manage College Information, Event Management, View Registrations, and Post Announcements—were built first, establishing the data foundation. Student modules—View College Information, View Event Calendar, Register for Events, and Chatbot Support—were developed subsequently and integrated with the completed admin back end. Dialogflow intents were trained on a corpus of representative student queries covering admissions, events, timetables, campus navigation, and contact details.

5) Deployment

After local development and testing, the application was deployed to a web hosting environment accessible via standard browsers. Google Sheets and Dialogflow are consumed as managed cloud services, eliminating the need for a dedicated database or NLP infrastructure.

V. RESULTS AND DISCUSSION

A. Functional Test Results

System testing covered Unit Testing (each module tested in isolation), Integration Testing (end-to-end data flow from form submission through Flask to Google Sheets), and Validation Testing (overall compliance with stated requirements). All five planned test cases passed:

Table 1: System Test Case Results

TC	Description	Test Data	Expected Result	Outcome
TC-01	Event registration form submission	Valid name, email, phone	Data stored in Google Sheets; confirmation email sent	Pass
TC-02	Admin login to management panel	Valid admin credentials	Login successful; dashboard displayed	Pass
TC-03	Chatbot response to college query	"What are the admission dates?"	Bot returns relevant answer	Pass
TC-04	Form validation on empty submission	Blank form	Validation errors shown	Pass
TC-05	Registration when capacity is full	Register when slots = 0	Registration blocked; message shown	Pass

B. Performance Analysis

The integration of the Dialogflow chatbot reduced the volume of routine administrative queries directed at staff by providing instant, accurate responses to the most frequently asked questions. Event registration data was transmitted reliably from the web form to Google Sheets across all test scenarios, with confirmation emails dispatched automatically upon each successful submission.

C. Discussion

The proposed portal successfully addresses all three problem areas identified during requirements analysis. Institutional information is now maintained centrally and propagated to the live site immediately upon administrator update. Event registration is fully automated, with capacity enforcement preventing over-subscription. The Dialogflow chatbot delivers 24/7 query resolution without staff involvement for routine requests.

The cloud-based architecture eliminates the need for dedicated server infrastructure, reducing the total cost of ownership. Role-based authentication ensures that only authorised administrators can modify portal content, preserving data integrity. Responsive design extends accessibility to mobile devices, broadening the portal's reach to students who primarily access the internet via smartphones.

VI. CONCLUSION

This paper presented the design and implementation of an Integrated College Web Portal with Event Management and Chatbot Support for SFRC, Sivakasi. The system consolidates institutional content management, automated event registration, and 24/7 student query resolution into a single, cost-effective platform built on Python Flask, Google Sheets, and Google Dialogflow.

Experimental results confirm that the system meets all stated functional and non-functional requirements. The modular, cloud-native architecture simplifies maintenance and supports future expansion. Planned enhancements include online fee payment integration, push notification support, advanced analytics dashboards for administrators, and expansion of Dialogflow intents to cover a broader range of student queries.

The proposed system demonstrates that modern web technologies and conversational AI can be combined effectively to deliver intelligent digital infrastructure for academic institutions, improving operational efficiency and the overall student experience.

REFERENCES

- [1] D. Gupta and B. Denton, "Appointment Scheduling in Web Portals: Challenges and Opportunities," IIE Transactions, 2008.

- [2] K. Batko, "The Use of Cloud Analytics and Google Workspace in Web Application Development," Journal of Big Data, 2022.
- [3] L. Li, "Development of College Information Systems and Smart Web Portals," Health Care Science, 2022.
- [4] M. H. Alrefaei and A. Diabat, "Modelling and Optimization of Web Portal Systems," RAIRO Operations Research, 2015.
- [5] Google LLC, "Dialogflow Documentation," [Online]. Available: <https://cloud.google.com/dialogflow/docs>
- [6] Pallets Projects, "Flask Documentation," [Online]. Available: <https://flask.palletsprojects.com>