IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

Campus Attendance Management System

Mr. Sayantan Debnath and Mrs. Swati Patil
Student, Department of Computer Technology
Guide, Department of Computer Technology
Bharati Vidyapeeth Institute of Technology Kharghar, Navi Mumbai, Maharashtra, India.

Abstract: *The Campus Attendance Management System is a digital solution designed to automate and streamline the process of recording, tracking, and managing student attendance in educational institutions. Traditional attendance methods are often time-consuming, error-prone, and difficult to maintain. This project aims to develop an efficient, accurate, and user-friendly system that replaces manual processes with a centralized digital platform.*
The system allows faculty members to mark attendance electronically through a web or mobile interface, while students can view their attendance records in real time. It incorporates features such as secure login authentication, course-wise attendance tracking, report generation, and data analytics to monitor student participation and performance. Advanced implementations may include biometric authentication, RFID technology, or QR code scanning to enhance accuracy and prevent proxy attendance..

Keywords: Attendance Management System, Campus Automation, Student Attendance Tracking, Digital Attendance, Web-Based Application, Mobile Application, Database Management System (DBMS), User Authentication, Real-Time Monitoring, Cloud-Based System

I. INTRODUCTION

In the modern era of digital transformation, educational institutions are increasingly adopting technology-driven solutions to automate and streamline administrative processes. One of the most critical and time-consuming tasks in any academic setting is the management of student attendance. Traditional manual attendance marking is prone to errors, proxy attendance fraud, and requires significant time from both teachers and students. To address these challenges, this project presents the B.V.I.T. Smart Attendance System — an AI-powered, web-based platform designed specifically for Bharati Vidyapeeth Institute of Technology, Navi Mumbai.

The main objective of this project is to transform the conventional paper-based or manual roll-call attendance system into an intelligent, automated experience using real-time face recognition technology. The system leverages the face-api.js library — a JavaScript implementation of deep learning face recognition models — to detect and identify students through a standard webcam, entirely within the browser without any server-side processing.

II. LITERATURE SURVEY

With the rapid advancement of Artificial Intelligence and computer vision technologies, automated attendance systems based on face recognition have gained significant attention in educational and corporate environments. This literature survey reviews existing research and technologies related to AI-based face detection, recognition systems, browser-based machine learning, and timetable-integrated attendance platforms, which form the foundational concepts of the proposed B.V.I.T. Smart Attendance System.

Research in deep learning-based face recognition has demonstrated that convolutional neural networks (CNNs) can achieve high accuracy in identifying individuals from visual data. Models such as FaceNet, developed by Schroff et al., use a triplet loss function to embed face images into a compact Euclidean space where distances correspond directly to face similarity. This concept is fundamental to the face recognition approach used in face-api.js, the JavaScript library integrated in the proposed system..

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

III. EXISTING SYSTEM

The current attendance management process in most educational institutions includes:

- Manual attendance marking using registers or spreadsheets
- Subject selection done manually by faculty
- Maintaining records on paper or basic digital files
- Lack of real-time monitoring and automation

Advantages of the proposed system:

- Automated face-based student identification using AI
- Real-time attendance marking with subject awareness
- Integrated live timetable for accurate class tracking
- Centralized dashboard with instant data updates
- Modular and scalable web-based architecture
- Eliminates proxy attendance and reduces human error

IV. PROPOSED SYSTEM

The B.V.I.T. Smart Attendance System proposes an intelligent, browser-based platform that automates attendance marking using AI face recognition. The system allows each student to register their face through a guided training session, capturing up to 50 face samples that are stored locally. During class, students scan their face using the webcam, and the AI model identifies them in real time. The system cross-references the current time with a built-in timetable to determine the active subject, ensuring that attendance is recorded for the correct class period. The entire process is managed through a modern web interface accessible from any device running a local server.

V. SYSTEM ARCHITECTURE

The B.V.I.T. Smart Attendance System follows a client-side, single-page architecture with no backend server. All processing — face detection, matching, data storage, and timetable logic — runs entirely in the user's browser. The system is organized into three logical layers:

The Presentation Layer is built with HTML5 (index.html) and CSS3 (style.css), providing the visual interface including the scanner panel, timetable card, student list, and modals. It uses the DM Sans and IBM Plex Mono fonts with a deep navy-blue color scheme aligned with B.V.I.T. branding.

The Logic Layer consists of four JavaScript modules: timetable.js manages schedule data and period detection; camera.js handles webcam access, video streaming, and real-time AI detection; faceai.js loads face-api.js models and provides face detection, matching, and registration functions; and app.js orchestrates all UI interactions including rendering, modals, toasts, and student management.

The Data Layer uses the browser's local Storage API to persist student records, face descriptors, and attendance configuration across page refreshes. The data.js file manages all CRUD operations on student records and serializes face descriptor arrays for storage.

VI. METHODOLOGY

The development lifecycle consists of the following phases:

Requirement Gathering and Analysis — Identifying the needs of TYCM2 students and faculty, including face recognition, timetable integration, and student management.

System Design — Planning the module structure, data flow, and UI layout across the six JavaScript and CSS files.

Module-wise Development — Building each file (data.js, faceai.js, camera.js, timetable.js, app.js, style.css) independently.

AI Model Integration — Loading and configuring face-api.js models from CDN for real-time face detection and recognition.

Data Persistence — Implementing local Storage for storing student records and face descriptors across sessions.

User Interface Design — Designing the dark navy-blue themed responsive interface consistent with B.V.I.T. branding.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

Testing and Validation — Testing face detection accuracy, timetable period detection, and student management workflows.

VII. RESULTS

Implemented system successfully:

The system successfully performs real-time face recognition to identify students accurately and mark attendance automatically for the active subject.

A centralized dashboard provides live statistics including total students, present, absent, and on-campus count, ensuring quick monitoring and transparency.

The student management panel dynamically updates attendance status, AI training status, and subject-wise records instantly after each successful detection.

The face registration module efficiently captures and trains multiple face samples with guided instructions, improving recognition accuracy and reliability.

The real-time timetable integration correctly detects the ongoing class, highlights the active period, and displays subject details with a live progress indicator.

The attendance confirmation workflow ensures accuracy through visual verification, confidence display, and instant updates with notifications, reducing errors and proxy attendance.

VIII. CONCLUSION

The B.V.I.T. Smart Attendance System was developed to overcome the limitations of manual attendance marking by providing an accurate, automated, and fraud-resistant solution using modern web technologies and browser-based Artificial Intelligence. It integrates real-time face detection and recognition through the face-api.js library, a live timetable synchronized with the TYCM2 (CM6K) schedule, and a comprehensive student management system within a modular JavaScript architecture. Students register their faces via a guided capture process, enabling high-confidence identification through a live webcam feed during class. A key feature is timetable-aware attendance marking, where the system automatically identifies the active subject based on real-time data, eliminating manual input and preventing incorrect entries. Its modular design ensures flexibility and maintainability, while local Storage enables backend-free operation with future scalability. Overall, the system effectively demonstrates the application of AI, computer vision, and web development in digitizing academic workflows, offering a reliable, efficient, and scalable attendance management solution.

REFERENCES

[1]. face-api.js by Vincent Mühler — The core library used for face detection, landmark detection, and recognition in the browser. GitHub:
[2]. F. Schroff, D. Kalenichenko, J. Philbin — "FaceNet: A Unified Embedding for Face Recognition and Clustering" — The foundational research paper behind the face recognition model architecture used by face-api.js. IEEE CVPR, 2015.
[3]. MDN Web Docs — "MediaDevices.getUserMedia()" — Official documentation for the browser API used to access the webcam stream.
[4]. MDN Web Docs — "Web Storage API (localStorage)" — Documentation for the localStorage API used to persist student records and face descriptors
[5]. MDN Web Docs — "HTML Canvas API" — Reference for the canvas element used to draw face bounding boxes and detection overlays on the video feed.
[6]. V. Kazemi and J. Sullivan — "One Millisecond Face Alignment with an Ensemble of Regression Trees" — Research behind the 68-point face landmark detection model used in face-api.js. IEEE CVPR, 2014.
[7]. TensorFlow.js Documentation — "Running Models in the Browser" — Reference for understanding browser-based model inference, which underpins how face-api.js executes neural networks client-side.
[8]. Google Fonts — IBM Plex Mono and DM Sans — Font resources used for the UI typography of the attendance dashboard.