OBE: Empowering the Future of Education

Suchitra Sona Gangurde[1], Mitali Vishnu Pawar[2], Sanket Shirish Chaudhary[3], Swetank Vinod Sonawane[4], Prof. M. A. Khan[5].

[1, 2, 3, 4] *Final Year Students, Department of Information Technology, Sandip Polytechnic, Nashik, Maharashtra, India*
[5] *Lecturer, Department of Information Technology, Sandip Polytechnic, Nashik, Maharashtra, India*

---***---

Abstract – *Outcome-Based Education (OBE) focuses on achieving predefined learning outcomes through structured teaching, learning, and evaluation practices. However, the manual documentation and tracking of these activities can be inefficient. This paper presents the design and development of a web-based application to automate the implementation of OBE under "Criteria 2: Teaching–Learning and Evaluation" for the Information Technology Department. The developed system facilitates digital curriculum planning, aligns Program Outcomes (POs) and Program Specific Outcomes (PSOs), and systematically documents academic activities. Key features include modules for managing expert lectures, industrial visits, value-added programs, and final-year projects. By utilizing a full-stack web framework, the system ensures transparency, improved student engagement, and real-time monitoring of educational goals. The outcomes indicate that digitizing these processes leads to better curriculum delivery and more effective attainment of learning outcomes.*

Key Words: **Outcome-Based Education, OBE, Web Application, Teaching–Learning Process, Curriculum Mapping, Academic Automation, MERN Stack.**

1. INTRODUCTION

Outcome-Based Education (OBE) has become a significant framework in technical education due to its focus on measurable learning outcomes and student-centered learning. While teaching, learning, and evaluation are crucial for academic effectiveness, manual methods of tracking these activities often result in data redundancy and a lack of transparency.

To address these challenges, we have developed a comprehensive software solution designed to digitize the "Criteria 2" standards. This paper discusses the development of this web application, highlighting its role in streamlining academic and co-curricular initiatives within the Information Technology Department to enhance learning quality and outcome attainment.

2. PROBLEM STATEMENT

Although various academic and co-curricular activities are conducted in polytechnic institutions, the absence of a structured, centralized digital framework for documentation limits their visibility and effectiveness. Manual record- keeping makes it difficult to map activities to specific Program Outcomes (POs). There is a critical need for a **systematic, web-based approach** to plan, monitor, and document teaching–learning activities in alignment with OBE principles to ensure continuous academic improvement.

3. OBJECTIVES AND SCOPE

3.1 Objectives:

- To design a software platform for the effective implementation of Outcome-Based Education practices.
- To digitally align curriculum delivery with Program Outcomes (POs) and PSOs.
- To improve transparency, accessibility, and documentation of academic activities through automation.

3.2 Scope:

The scope of this project includes the digitization of curriculum planning, teaching–learning processes, expert lectures, industrial visits, value-added programs, industrial training, and final-year projects under the umbrella of Criteria 2.

4. LITERATURE REVIEW

Outcome-Based Education (OBE) has been widely adopted in technical and higher education to improve learning effectiveness and ensure measurable academic outcomes. Academic frameworks and accreditation bodies emphasize aligning curriculum design, teaching–learning activities, and assessment methods with clearly defined Program Outcomes (POs) and Program Specific Outcomes (PSOs). Studies indicate that structured outcome mapping enhances curriculum transparency and supports continuous quality improvement in educational institutions.

However, several researchers have highlighted challenges in the practical implementation of OBE, particularly when academic activities are documented using manual or semi- digital methods. Manual record-keeping increases administrative workload, limits real-time monitoring, and makes it difficult to retrieve evidence during academic audits. Recent literature suggests that integrating information technology solutions into academic management systems can significantly improve data accuracy, accessibility, and documentation efficiency.

Existing studies on academic automation and web-based management systems demonstrate that digital platforms support centralized data storage, role-based access, and automated report generation. These systems reduce redundancy and improve institutional readiness for accreditation and quality assurance processes. Based on these findings, this study focuses on developing a web-based solution to support the effective and practical implementation of OBE under Criteria 2.

5. PROPOSED SYSTEM / METHODOLOGY

The proposed system is a Full-Stack Web Application designed to organize, monitor, and document all teaching– learning activities. The system architecture enables the mapping of curriculum content with Program Outcomes and PSOs to ensure strict outcome alignment.

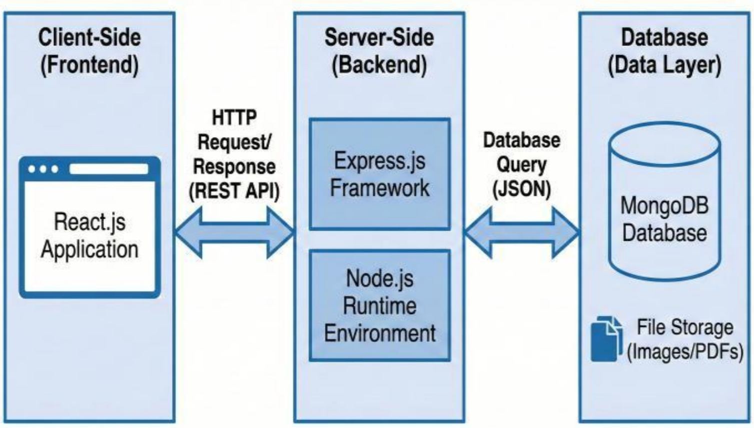

Fig -1: Proposed System Architecture (MERN Stack)

5.1 Module Description

The proposed web application is divided into three primary functional modules to ensure secure and role-based access:

A. Administrator Module: The administrator has the highest level of privilege. The key responsibilities of this module include:

- **User Management:** Creating and managing accounts for Heads of Departments (HOD), faculty members, and students.
- **Curriculum Mapping:** Defining the Program Outcomes (POs) and Program Specific Outcomes (PSOs) for the academic year.
- **System Monitoring:** Viewing overall usage statistics and ensuring the system is running correctly.

B. Faculty / Staff Module: This is the core module where the majority of data entry occurs. Faculty members can:

- **Activity Upload:** Select the type of activity (e.g., "Guest Lecture" or "Industrial Visit") and upload relevant details such as date, resource person, and photographs.

- **Outcome Mapping:** directly maps a specific activity to the relevant POs (e.g., mapping an "Industrial Visit" to "PO-2: Engineering Knowledge").
- **Report Generation:** Generate automated PDF reports for their specific subjects or criteria files.

C. Student Module (View Only): Students are provided with read-only access to transparency. They can:

- **View Calendar:** Check the schedule of upcoming academic activities.
- **Access Resources:** Download study materials or reports uploaded by faculty.
- **Feedback System:** Submit feedback regarding specific events or lectures, which is stored in the database for quality improvement analysis.

5.2 Report Generation Algorithm

One of the critical features of the system is the automated generation of documentation. The backend utilizes a specific algorithm to fetch data and convert it into a standardized format:

1. **Fetch Request:** The user selects the "Generate Report" button for a specific date range.
2. **Data Retrieval:** The Node.js server queries the MongoDB database for all activities matching the filter.
3. **Template Mapping:** The retrieved JSON data is mapped into a pre-defined HTML template.
4. **PDF Conversion:** A library (such as Puppeteer or PDFKit) converts the HTML view into a downloadable PDF file.
5. **Download:** The file is sent to the client's browser for the user to save locally.

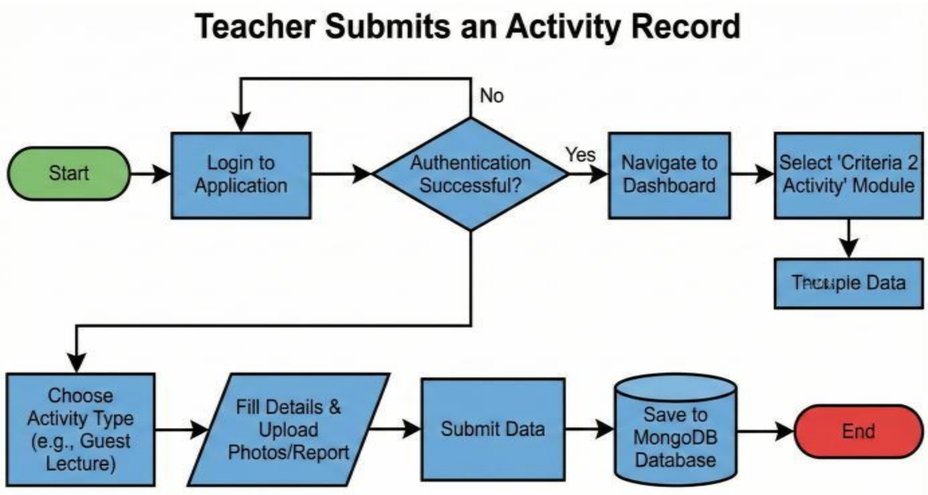

Fig -2: Flowchart for Activity Data Submission

6. TOOLS AND TECHNOLOGIES

The proposed system is developed using the **MERN Stack** (MongoDB, Express.js, React.js, Node.js), selected for its scalability, efficiency, and full-stack JavaScript capability. To ensure code quality and a modern user interface, the development process also integrated **TypeScript** for type safety and **Tailwind CSS** for responsive styling.

6.1 Frontend Technologies

The client-side of the application is built to be responsive and intuitive, ensuring seamless access for both students and faculty.

- **React.js:** A component-based JavaScript library used for building the user interface. It allows for the creation of reusable UI components, ensuring a consistent look and feel across the application.
- **Tailwind CSS:** A utility-first CSS framework used for rapid UI development. It allows for highly customizable designs without writing standard CSS, ensuring the application is fully responsive on mobile and desktop devices.
- **TypeScript:** Used as a superset of JavaScript to introduce static typing. This reduces runtime errors and improves code maintainability, which is critical for handling complex logic in the outcome mapping modules.

6.2 Backend Technologies

The server-side logic is designed to handle multiple concurrent requests and manage secure data transactions.

- **Node.js:** A JavaScript runtime environment that executes code server-side. Its non-blocking I/O model makes it ideal for handling real-time data fetching in this application.
- **Express.js:** A minimal web application framework for Node.js. It is used to build robust RESTful APIs that facilitate communication between the React frontend and the MongoDB database.

6.3 Database Management

- **MongoDB:** A NoSQL database used to store unstructured data such as student records, activity reports, and images. Unlike traditional SQL databases, MongoDB stores data in JSON-like documents, which allows for flexible schema design—perfect for the varying data structures of different academic activities.

6.4 Development Tools

- **Git & GitHub:** Used for version control to track changes in the source code and facilitate collaboration among team members.
- **Postman:** Utilised for testing API endpoints during the development phase to ensure that data requests and responses are functioning correctly before connecting them to the frontend.
- **VS Code:** The primary Integrated Development Environment (IDE) used for writing and debugging code.

7. EXPECTED OUTCOMES

The implementation of the proposed web-based management system is expected to yield significant improvements in the administrative and academic workflows of the department. The key anticipated outcomes include:

- **Automated Curriculum Alignment:** The system ensures seamless mapping of every academic activity with specific Program Outcomes (POs) and Program Specific Outcomes (PSOs), eliminating manual errors in attainment calculations.
- **Enhanced Student Participation:** By increasing the visibility of industry-oriented activities through the student portal, the system encourages higher student engagement in workshops, expert lectures, and value-added programs.
- **Digital Transparency & Accessibility**: A centralized cloud-based platform ensures that all academic documentation is transparent, secure, and easily retrievable by stakeholders from any location, resolving the issue of physical file dependency.
- **Data-Driven Quality Assurance:** The system provides real-time analytics and graphical dashboards, enabling the Head of Department (HOD) and faculty to make data-driven decisions to strengthen academic quality and facilitate continuous improvement.
- **Audit Readiness**: With automated report generation, the department will remain perpetually ready for external academic audits (such as NBA or NAAC) with organized, standard-compliant digital records available on demand.
- **Resource Optimizations:** The transition from paper-based logs to a digital database will significantly reduce paper usage and administrative man-hours, allowing faculty to focus more on teaching rather than clerical documentation.

Table -1: Comparison of Manual System and Proposed Web System

Features	Manual System	Proposed Web System
Data Handling	Physical files & registers	Digital Database (MongoDB)
Accessibility	Restricted to the Office	Accessible anywhere
Report Generation	Manual, Time-consuming	One–click PDF

Transparency	Difficult to track	Real-time Dashboard
Outcome Mapping	Difficult	Automated & Accurate
Data Security	High Risk of Data Loss	Secure Authentication & Backup

8. RESULT AND IMPLEMENTATION:

The proposed web-based Outcome-Based Education management system was successfully implemented at the departmental level to support the documentation and monitoring of *Criteria 2: Teaching–Learning and Evaluation*. The system was deployed in a controlled academic environment and tested using actual departmental activities and records.

Faculty members accessed the system through role-based authentication to upload details of academic and co- curricular activities such as expert lectures, industrial visits, value-added programs, and final-year projects. Each activity was mapped to relevant Program Outcomes (POs) and Program Specific Outcomes (PSOs) using predefined options, ensuring consistent outcome alignment. Supporting documents, including reports and photographs, were stored digitally in the database.

The administrator module enabled centralised management of users, academic parameters, and outcome definitions, while automated report generation allowed faculty to produce standardised PDF reports for documentation and audit purposes. Students were provided with view-only access to schedules and shared resources, improving transparency and awareness of academic activities.

The implementation demonstrated improved efficiency in academic documentation, reduced manual effort, and enhanced accessibility of records. The use of screenshots (Fig -3 to Fig -7) illustrates key system interfaces, validating the practical functionality and usability of the proposed solution.

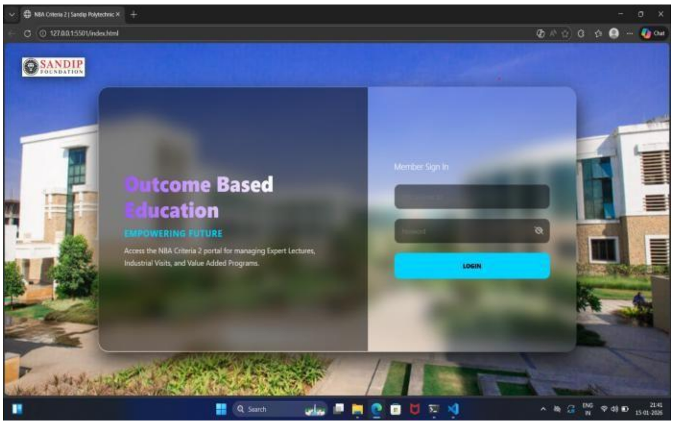

Fig -3: Administrator Dashboard Interface

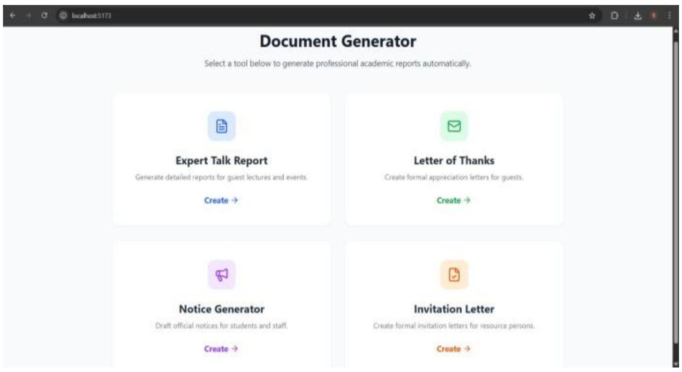

Fig -4: Faculty Activity Upload Module

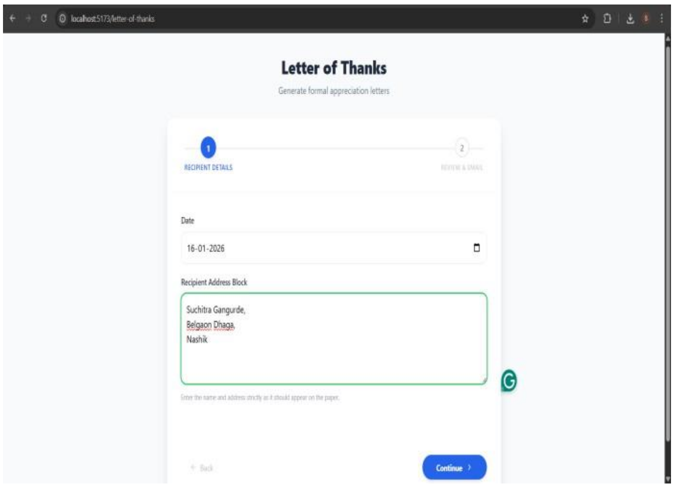

Fig -5: Program Outcome Mapping Screen

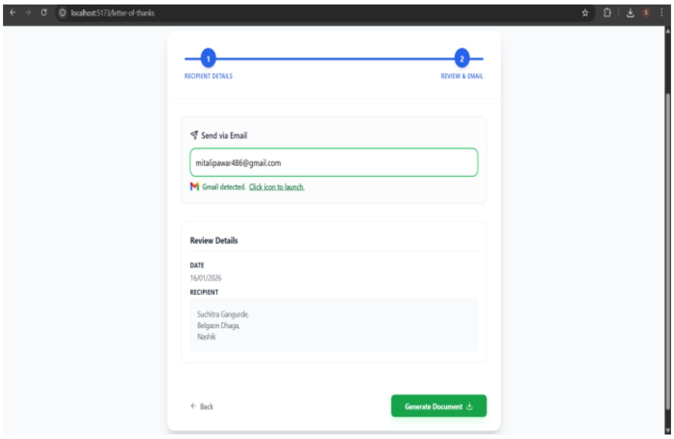

Fig -6: Generated PDF Report

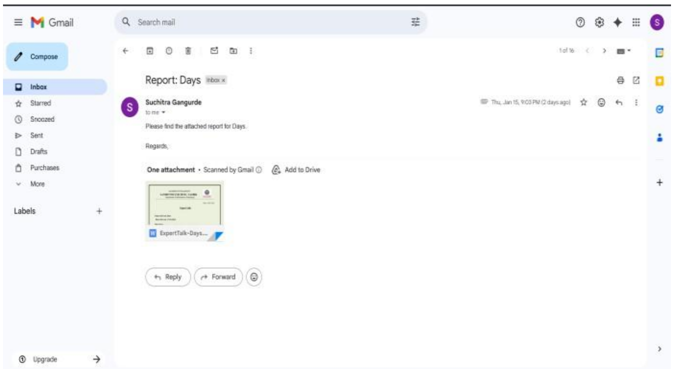

Fig -7: Sending Report to the Recipient Automatically

9. FUTURE SCOPE:

While the current system effectively digitizes the "Criteria 2" documentation, there is significant potential for future enhancements:

- AI-Based Analysis: Implementing Machine Learning algorithms to predict student performance trends based on the mapped outcomes.
- Mobile Application: Developing a dedicated Android/iOS app using React Native for easier access by faculty on the go.

- Integration with ERP: The system can be integrated with the college's existing ERP software to fetch student attendance data automatically.
- Blockchain Security: Using blockchain technology to issue tamper-proof digital certificates for students who complete specific value-added programs.

10. CONCLUSION

The development and implementation of this web-based OBE management system demonstrates a significant improvement in the effectiveness of teaching–learning processes. By shifting from manual to digital documentation, the department can ensure better curriculum alignment and continuous monitoring. This study confirms that integrating modern IT solutions with educational frameworks supports quality assurance and institutional growth in technical education.

ACKNOWLEDGEMENT

The authors express sincere gratitude to our project guide, Prof. M.A. Khan, and the faculty members of the Department of Information Technology for their continuous guidance and support. Special thanks are extended to **Sandip Foundation's Sandip Polytechnic** for providing the necessary resources and infrastructure to complete this research.

REFERENCES

[1] MSBTE Academic Calendar, Maharashtra State Board of Technical Education.

[2] National Board of Accreditation (NBA), Outcome-Based Education Guidelines.

[3] IEEE Publications on Outcome-Based Education and Academic Quality Assurance.

[4] Departmental Reports & COE Records, Sandip Polytechnic.

[5] W. G. Spady, "Outcome-Based Education: Critical Issues and Answers," American Association of School Administrators, Arlington, VA, 1994.

[6] R. S. Pressman, "Software Engineering: A Practitioner's

Approach," 7th ed., McGraw-Hill Higher Education, 2010.

[7] V. D. S. Silva, "The MERN Stack: A Review of the Learning Curve and Application in Modern Web Development," International Journal of Computer Applications, vol. 176, no. 3, pp. 25-30, 2020.

[8] "React - A JavaScript library for building user interfaces," Facebook Open Source.

[9] "MongoDB Documentation," MongoDB Inc.

[10] S. Gupta and M. Kumar, "Automation of Academic Management Systems in Higher Education," IEEE International Conference on Advanced Computing, 2019.

[11] J. Dean, "Web Application Architecture: Principles, Protocols and Practices," Wiley Publishing, 2009.

[12] "Express.js: Fast, unopinionated, minimalist web framework for Node.js," OpenJS Foundation.