



Jaypee University of Information Technology Department of Computer Science and Engineering and Information Technology

Major Project - II (18B19CI891) | AY 2024-25

End-Term Evaluation | May 16-17, 2025.

Student Behaviour Management

Group No.: 121

Team Member (s)

- Manas Bajpai (211477)
- Designation: Frontend Intern
- Organization: Toddle
- Location: Remote

Supervisor (s)

Name: Ms. Seema Rani

Department: CSE & IT

Name: Mr. Gaurav Negi

Department: CSE & IT

Outline



- Introduction
- Problem Statement
- Objectives
- Work Done (after Mid-Term Evaluation)
- Project Design
- Implementation
- Experimental Results and Evaluation
- Key Learnings
- Future Work
- Work Contribution and Attendance
- Supervisor Interactions
- References

Introduction



- Student Behaviour Management is a tool designed to help schools document and monitor all types of student behaviour — whether positive, neutral, or negative.
- The system provides a centralized platform where teachers can log incidents such as achievements, participation, misconduct, or bullying, ensuring every behavioural event is traceable.
- By streamlining behaviour tracking, the module aims to foster accountability, data-driven decision-making, and holistic student development.
- The module also helps build a transparent record of student conduct over time, supporting more informed parent-teacher discussions and disciplinary decisions.
- Additionally, it fosters a culture of accountability and recognition by ensuring both positive and negative behaviours are consistently acknowledged.

Problem Statement



Most schools lack a structured and collaborative system to track student behaviour.

As a result:

- Minor yet significant incidents often go unreported.
- There's no consistent way to assess behavioural trends.
- Teachers struggle to coordinate effectively when managing student conduct.

This leads to missed opportunities for both recognition and early intervention.

Objectives



The module aims to streamline and enhance the process of managing student behaviour through the following core objectives:

1. Centralized Reporting System

Provide a unified platform where all student behaviour incidents are logged in real-time, ensuring nothing slips through the cracks.

2. Collaborative Incident Handling

Enable collaboration by allowing teachers to tag other staff and students involved in an incident — supporting cases that require input from multiple parties.

3. Referral Workflow for Interventions

Allow teachers to escalate or refer incidents to other faculty members or school counsellors for appropriate follow-up or intervention.

4. Clear Incident Lifecycle Management

Introduce a three-stage status system to ensure transparency and accountability:

- Draft: Incident is being created and may need edits.
- Published: Incident is active and awaiting actions from involved parties.
- **Resolved**: Incident is closed and archived no further changes allowed.

Work Done (after Mid-Term Evaluation)



Teacher Notes in Student Profile

- Added a section where any teacher can log notes related to a student.
- Each note includes a title, type, description, and optionally attachments.
- These notes can cover observations, reminders, or any context that may help in understanding the student better.

Student Alert Types in School Settings

- Introduced a configurable system where admins can define custom alert types (e.g., medical condition, learning disability, disciplinary flag).
- Enables flexible alert tagging across various school contexts.

Student Alerts Tab in Student Profile

- Implemented a dedicated section for adding, editing, or removing **student alerts**.
- Alerts include a label, description, and optionally attachments.
- Visually represented as icons beside the student's name throughout the system, acting as quick flags to inform staff decisions.

Project Design



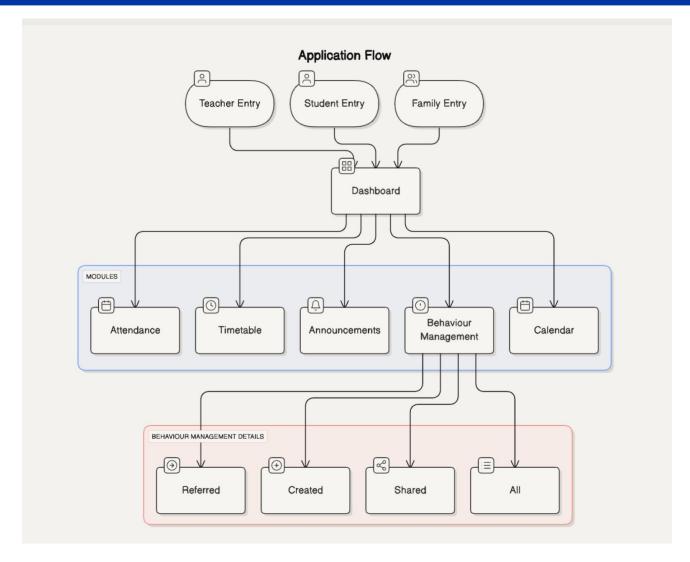


Fig 1: Flowchart of the application, emphasised on the Behaviour module

Project Design (cont...)



User Identification:

On launch, the system asks the user type before logging him into the system. Users choose from **Parent**, **Teacher**, or **Student**.

Login Flow Based on Role:

- Teachers log in via email, Google, or other supported socials.
- Parents/Students use unique login codes provided by the school.

Role-Based Module Access:

After login, a JWT is issued containing the user's role. The system then displays only the relevant modules — including the Behaviour Module with tailored features for each role.

Implementation



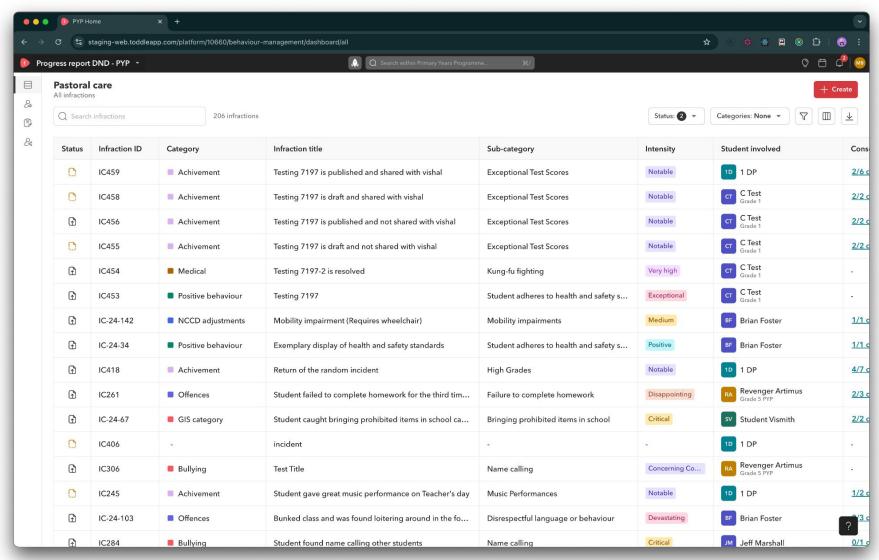


Fig 2: Illustration of the behaviour module's homepage showcasing all active incidents.

Implementation (cont...)



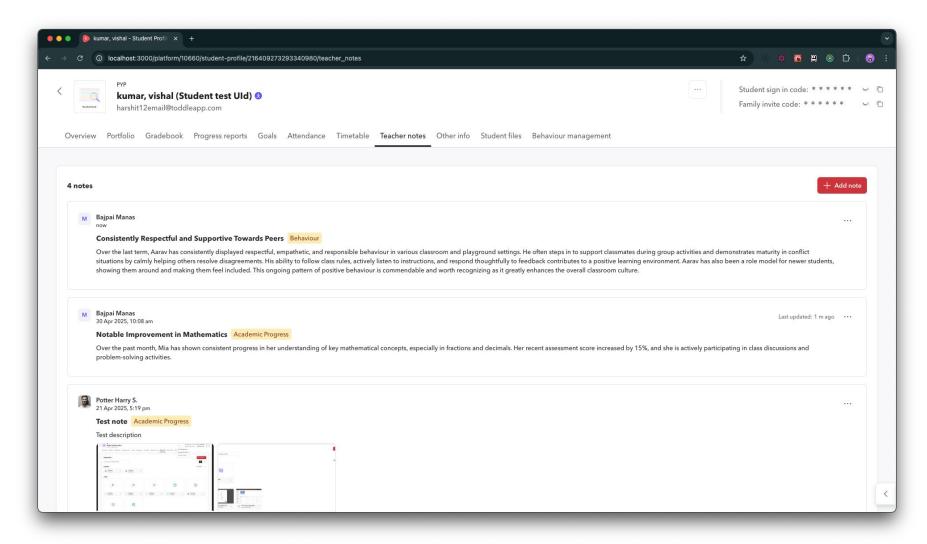


Fig 3: An illustration of the teacher notes section in the student profile

Implementation (cont...)



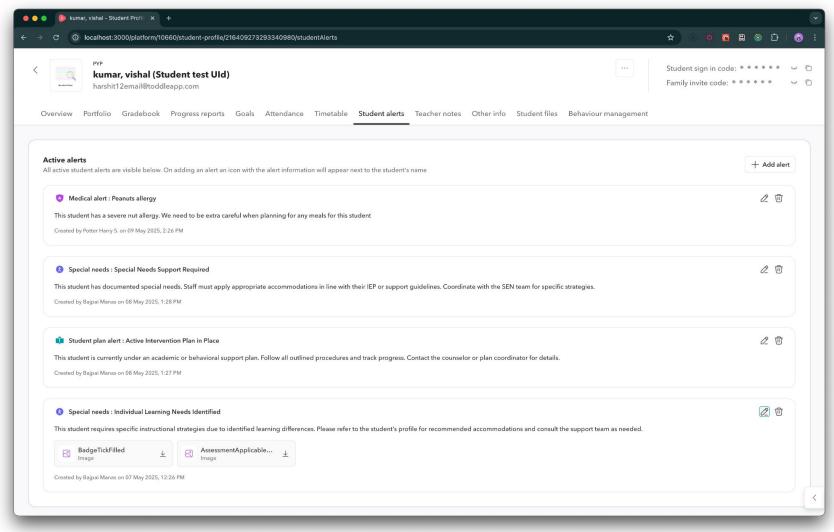


Fig 4: An illustration of the student alerts page

Experimental Results and Evaluation



Multi-Environment Testing

- The system was tested in development, staging, and production environments.
- It consistently maintained an average response time of under 400 ms for both logging and retrieving incidents.

High Concurrency Handling

- No performance degradation was observed during peak usage.
- The system handled simultaneous inputs from multiple users including teachers, students, and family members.

Real-Time Reporting Accuracy

- The system achieved 100% accuracy in reflecting real-time changes across all user interfaces.
- Optimistic UI updates were tested across overlapping incident scenarios.

Accessibility Compliance

- All accessibility features were validated against WCAG 2.1 standards.
- Keyboard navigation and shortcuts worked across major devices and browsers.

Experimental Results and Evaluation (cont...)



Data Quality and Consistency

- Automated data validation checks were implemented to ensure clean records.
- These covered edge cases like incomplete entries and cross-module inconsistencies.

User Feedback

- Initial feedback from educators and staff highlighted the platform's ease of use.
- Tagging and referral features were especially appreciated for simplifying collaborative workflows.

Confidential Notes

- The confidential notes section was well received by users.
- It allowed teachers to document sensitive observations without affecting the main incident flow.

Accessibility Compliance

- All accessibility features were validated against WCAG 2.1 standards.
- Keyboard navigation and shortcuts worked across major devices and browsers.

Experimental Results and Evaluation (cont...)



Student Alerts

- The system now sends alerts for repeated negative behavior or serious incidents.
- This enables timely intervention by teachers and administrators.

Teacher Notes

- Teachers can add private notes for additional context on student behavior.
- These notes are only visible to authorized users and support better decision-making.

Overall Observations

- Validation mechanisms and improved navigation significantly enhanced the user experience.
- Users reported higher accuracy and ease of use across all roles.

Key Learnings



- Gained experience in building scalable, real-time applications using React, Apollo Client, and GraphQL.
- Learnt how to write code in a clean and efficient way so that it is well-readable and maintainable for a long-term.
- Learned to work on a product that is in production and has active users with frequent new feature requests from them (i.e. schools)
- Learned version management, how to handle and debug bugs that are normally unseen and are caught under very edge-case conditions.
- Learned how to design systems that support multi-user collaboration in an educational environment.
- Improved knowledge of working with GraphQL using Apollo Client for managing complex API calls.

Future Work



- Will work on revamping my module as per Toddle 2.0, which is a complete revamp of the application.
- Will work on fixing more bugs throughout the behaviour module and profiles to make the user experience better.
- Will work on completing student alerts integration across the entire application, so that teachers are able to identify students easily and make decisions based on relevant data.

Work Contribution and Attendance



GitHub Repository URL: https://github.com/MacWeTT/StudentBehaviourManagement

Team Member	Roll No.	Work Done (provide complete details)	Work Contribution (%)	Lines of Code (LoC)	Lab Attendance (%)
1.	211477	Worked on the behaviour management module.	100	12000+	-

Supervisor Interactions (as mentioned in weekly log)



No. of Meetings with Supervisor:

Week No.	Duration	Remarks (as mentioned in the weekly log)	Incorporated (Yes/No)				
1.	24/03/2025 to 29/03/2025	 Add missing test-ids as observed in the student profile. 	Yes				
2.	31/03/2025 to 05/04/2025	Implement DQC changes as described by the product designer.	Yes				
3.	07/04/2025 to 12/04/2025	 Optimise code logic, performance can be improved. Handle edge-cases in an efficient manner in future contributions. 	Yes				
4.	14/04/2025 to 19/04/2025	 Complete scoping carefully and ask any relevant doubts to the product designer so that issues are not faced during development. 	Yes				
5.	21/04/2025 to 26/04/2025	Try to reduce prop drilling and import libraries carefully.	Yes				

Supervisor Interactions (cont...)



Week No.	Duration	Remarks (as mentioned in the weekly log)	Incorporated (Yes/No)
6.	28/04/2025 to 03/05/2025	 Error states can be handled in a more efficient way. Can write logic in a cleaner way. 	Yes
7.	05/05/2025 to 10/05/2025	Fix any technical debt you can find in the codebase.	Yes
8.	12/05/2025 to 16/05/2025	-	-

References



- [1] ReactJS Documentation, React A JavaScript library for building user interfaces, 2024. [Online]. Available: https://react.dev/. [Accessed: Nov. 30, 2024].
- [2] Apollo GraphQL, *Apollo Client Documentation*, 2024. [Online]. Available: https://www.apollographql.com/docs/react/. [Accessed: Nov. 30, 2024].
- [3] PostgreSQL Documentation, *PostgreSQL: The World's Most Advanced Open Source Relational Database*, 2024. [Online]. Available: https://www.postgresql.org/. [Accessed: Nov. 30, 2024].
- [4] Web Content Accessibility Guidelines (WCAG), Web Content Accessibility Guidelines 2.1, W3C, 2024. [Online]. Available: https://www.w3.org/WAI/standards-guidelines/wcag/. [Accessed: Nov. 30, 2024].
- [5] Toddle Internal Documentation, Internal guidelines and system documentation, Proprietary Documentation, 2024.
- [6] Sentry Documentation, Sentry: Application Monitoring and Error Tracking Software, 2024. [Online]. Available: https://sentry.io/. [Accessed: Nov. 30, 2024].
- [7] Jira Documentation, *Jira Software by Atlassian*, 2024. [Online]. Available: https://www.atlassian.com/software/jira. [Accessed: Nov. 30, 2024].

References (cont...)



- [8] AWS Documentation, AWS Cloud Computing White Papers, "Amazon Web Services", 2024. [Online]. Available: https://aws.amazon.com/whitepapers/. [Accessed: Nov. 30, 2024].
- [9] Slack API Documentation, "Slack API: Build your integration, 2024". [Online]. Available: https://api.slack.com/. [Accessed: Nov. 30, 2024].
- [10] Lodash Documentation, "A modern JavaScript utility library delivering modularity, performance & extras, 2024". [Online]. Available: https://lodash.com/docs/. [Accessed: Nov. 30, 2024].
- [11] Ant Design, "Ant Design A design system for enterprise-level products, 2024". [Online]. Available: https://ant.design/docs/react/introduce. [Accessed: Nov. 30, 2024].
- [12] Redux Toolkit Documentation, "Redux Toolkit: The official, recommended approach for writing Redux logic, 2024". [Online]. Available: https://redux-toolkit.js.org/. [Accessed: Nov. 30, 2024].
- [13] React Router Documentation, "React Router: Declarative routing for React apps, 2024". [Online]. Available: https://reactrouter.com/. [Accessed: Nov. 30, 2024].
- [14] Tailwind CSS Documentation, *Tailwind CSS: A utility-first CSS framework for rapidly building custom user interfaces*, 2024. [Online]. Available: https://tailwindcss.com/docs. [Accessed: Nov. 30, 2024].



