

PROJECT REPORT

ENGAGETRACK

A Smart AI-Powered Class Engagement Monitoring System

CHOSEN PROBLEM STATEMENT: 4

TEAM MEMBERS:

- 1. MEHER SAI SANJANA POTUKUCHI (TEAM LEAD)**
- 2. POLAMARASETTI GOWTAM**

FACULTY MENTOR:

SURAJ ARAVIND B

UNIVERSITY NAME:

GITAM UNIVERSITY

Abstract:

Ensuring students' active engagement and comprehension is a recurring challenge in contemporary education. A comprehensive, AI-powered classroom engagement monitoring tool, EngageTrack is designed to help teachers assess students' participation and understanding in real time. The system gives lecturers and schools the ability to improve the quality of learning by incorporating modules for essay-based comprehension analysis, student engagement monitoring, and live lecture transcription. The platform, which was developed with FastAPI, MongoDB, JavaScript, HTML, and CSS, provides teachers and students with a smooth experience and user-friendly dashboards.

Introduction:

In a classroom environment, it can be difficult for teachers to keep an eye on every student's participation and comprehension in large or diverse classrooms. This frequently results in passive learning, where students may fall behind if proper action is not taken. To improve teaching and learning outcomes, EngageTrack combines artificial intelligence with real-time behavioural analytics.

Teachers can record lectures using the platform's voice input feature, and the lectures will be automatically transcribed. Through the analysis of keystroke patterns, typing speed, pause times, and focus shifts, it concurrently monitors student participation in typing-based activities. Additionally, EngageTrack offers essay-based comprehension analysis modules that reveal how well students comprehended the lecture. It provides seamless, intuitive dashboards for educators and learners alike, thanks to its construction with FastAPI, MongoDB, JavaScript, HTML, and CSS.

Objectives:

EngageTrack's main goal is to improve classroom instruction by utilising real-time data analytics and artificial intelligence. Its objectives are:

- To increase student engagement in the classroom with real-time AI-powered tools
- To help teachers quickly identify students who are struggling or disengaged
- To assess student understanding using an essay-based assessment
- To give educators a user-friendly and interactive dashboard
- To provide students with a customised dashboard that shows lecture transcripts, engagement data, and feedback for self-paced learning.

Work Division:

Meher Sai Sanjana Potukuchi:

The following AI modules were created and put into use:

- Sentence Transformers for essay-based comprehension analysis

The dashboards for teachers and students were created and designed.

Integrated Web Speech API for real-time lecture transcription.

FastAPI was used to create backend APIs, and HTML, CSS, and Jinja2 were used to create and style several frontend pages.

Handled project documentation, report writing, and final presentation.

Polamarasetti Gowtam:

The following AI modules were created and put into use:

- Random Forest model and typing behaviour for engagement tracking

Used MongoDB to implement database schemas and data storage logic.

Managed the submission of forms, handled sessions, and integrated backend logic

Helped with UI improvements and frontend design.

Carried out thorough feature testing and debugging throughout the stack.

Made a demonstration video to show how the project operates.

System Architecture:

The following crucial components were used to create EngageTrack, a complete full-stack application:

Frontend: Created with HTML, CSS, JavaScript, and Jinja2 templating, it offers user interfaces for submitting essays, interacting with students, and tracking engagement in real time.

Backend: Using FastAPI, it manages database interactions, routing, AI processing, and API logic.

Database: MongoDB holds class information, essays, typing logs, session information, and user data.

AI /ML services: Intelligent insights are produced through the integration of AI modules. The system employs Sentence Transformers for semantic comparison of essays with lecture content, a Random Forest model to analyse typing behaviour, and the Web Speech API for real-time lecture transcription. The system was further improved by a planned OpenVINO-optimized mini LLM model that will provide feedback and respond to student enquiries according to the context of the lecture.

Key Features:

Lecture Transcription: Lectures can be recorded in real time by teachers. Using the Web Speech API, speech is transformed into text and stored in the database.

Facilitates essay analysis and allows students to review material.

Student Engagement Tracking: Uses keyboard input logging to track activity, pause duration, and typing speed. Monitors whether students are engaged or not. Shows the teacher dashboard's engagement status (Engaged/Not Engaged) and if the essay typed is thoughtful or copied from another source.

Essay-Based Comprehension Analysis: After the lecture, students submit essays. Essays and the transcribed material are compared to find any points that are unclear, misunderstood, or missing. AI-generated feedback aids educators in evaluating students' comprehension.

Technologies Used:

LAYER	TOOLS/Frameworks
Frontend	HTML, CSS, JavaScript
Backend	FastAPI (Python)
Database	MongoDB
AI/ML Features	Web Speech API, Random Forest Model for Typing Analysis, OpenVino Optimized Mini LLM Model, and Sentence Transformers are used throughout the model.

Metrics Used:

1. Typing Behavior Metrics (Engagement Tracking):

Typing Speed (WPM): This metric is used to assess how actively and consistently a student types throughout a session.

Pause Duration (ms): Measures the amount of time that passes between keystrokes to detect inattention or disengagement.

Keystroke Frequency: This measure of active participation counts the number of keystrokes per minute.

Engagement Score: A Random Forest model is used to calculate this metric, which is based on speed, frequency, and pause time and is used to categorise students as either engaged or not(thoughtful/ copied).

2. Semantic Similarity Metrics (Essay Analysis):

Cosine Similarity Score: Sentence Transformers are used to generate the Cosine Similarity Score, which measures how closely a student's essay adheres to the lecture's transcribed content.

Coverage Ratio: The percentage of important lecture points that are included in the essay

Relevance Score: Assesses how well essay sentences relate to the lecture material.

3. System Usability Metrics:

Response Time: Backend response times for tracking engagement and submitting essays.

Transcription Accuracy: Transcription accuracy is qualitatively assessed by comparing the output of the Web Speech API with spoken content.

Implementation Details:

Backend API (backend/api):

lecturer.py: Manages endpoints for analysis, lecture retrieval, and recording

student.py: Manages engagement status, typing logs, and essay submissions.

auth.py: Controls user authentication and login.

Database Layer (backend/db/db.py):

Interfaces with typing_logs, classes, students, essays, sessions, and other collections.

Frontend:

Teachers can view real-time engagement analytics, and students can type essays, record speeches, and view results using the HTML files(both student_dashboard and lecturer_dashboard) under the frontend folder.

Main App Entry Point:

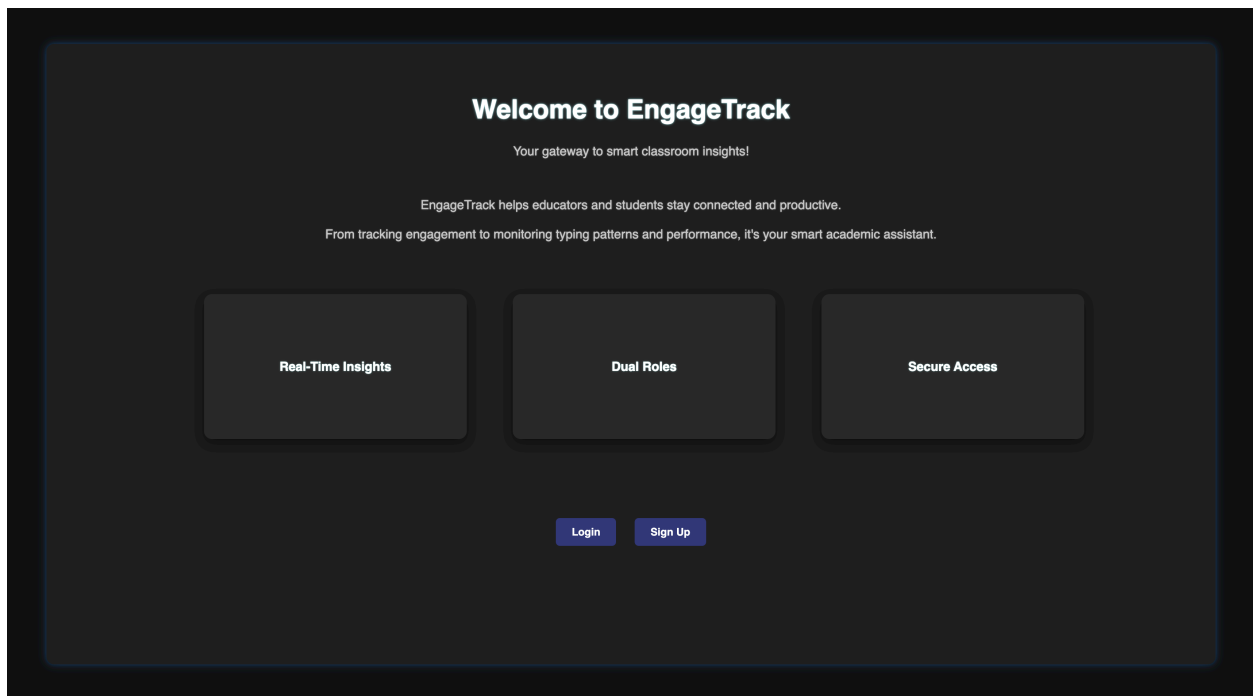
The FastAPI application is initialised, and all routers are mounted by main.py.

Speech Recognition Integration:

Uses the Web Speech API in the browser to record live audio and turn it into text transcription.

Screenshots of the project:

HOME PAGE:



LOGIN & SIGNUP PAGE:

Create an Account

Role:

Student

First Name:

Last Name:

Email:

Phone:

ID (Roll Number or Employee ID):

Password:

Sign Up

Already have an account? [Login here](#)

Login to EngageTrack

Role:

Student

ID (Roll No. / Employee ID):

Password:

Login

Don't have an account? [Sign up here](#)

LECTURER DASHBOARD:

Welcome, Gowtam!

Your Current Classes

Class ID: CSE101

Subject: Programming with c

Manage

Classes Done

No completed classes yet.

+ Add New Class

Logout

Add a New Class

Class ID:

CSE101

Subject:

Programming with c

Select Students to Enroll:

☐ Alice Student (S123)

☐ GOWTAM POLAMARASETTI (S1234)

☒ Gowtam P (g123)

☒ Meher Sanjana (L202)

☐ Kumar Polamarasetti (S03)

Create Class

Manage Class: Computer Science (CSE101)

Use the options below to manage student enrollment and sessions

👤 Current Students

Gowtam P (g123)

Remove

Meher Sanjana (L202)

Remove

✔ Update Student List

+ Add Students

Alice Student (S123)

GOWTAM POLAMARASETTI (S1234)

Kumar Polamarasetti (S03)

Add Selected Students

📺 Start Class Session

📹 Start Recording Session

Record Lecture for Class ID: 6859a12efe4bc7197d7b9225

Start Session

Record Lecture for Class ID: 6859a12efe4bc7197d7b9225

Start Session

Start Voice Recognition

Transcript (editable):

Voice transcript will appear here...

Submit Transcript

Analytics for Class: OS (CSE202)

1 Overall Engagement

Average Engagement Score: 100.0%


2 Students Who Submitted Essays

Gowtam P (g123)


View Engagement

3 Students Who Have Not Submitted

All students have submitted.

 [Back to Dashboard](#)

 Student: Gowtam P (g123)

 Class: OS (CSE202)

 Essay Submitted:

Os, also known as an Operating System, is a system software. It acts as an intermediary between the hardware and the software. It can be used in Memory allocation, process management and file handling

 Engagement Report:

Engagement Score: 1

Typing Style: thoughtful

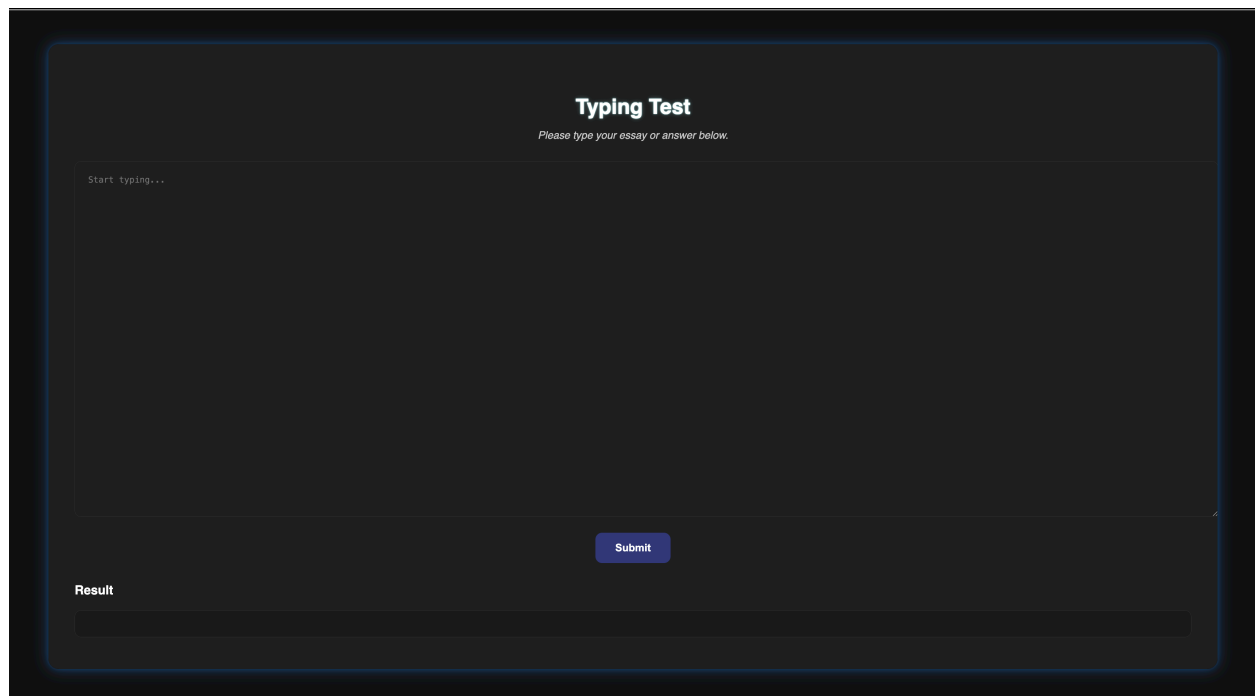
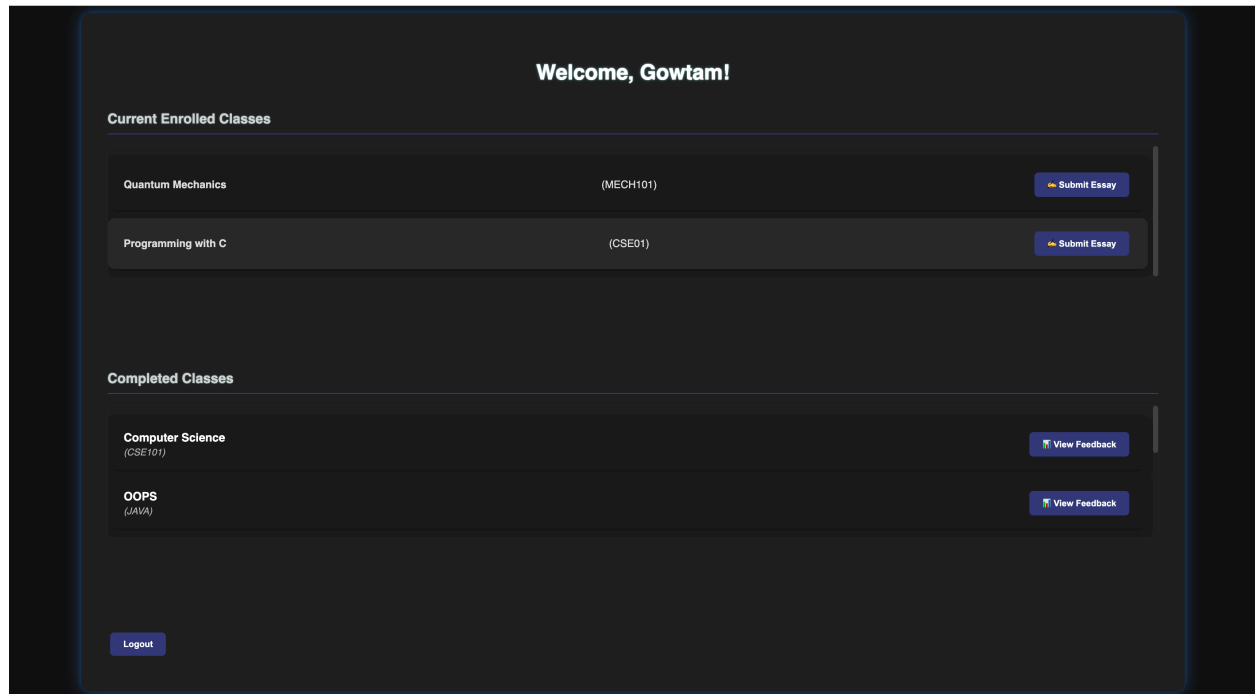
Similarity Score: 2.27

 Submit Feedback

Very good!!

 Save Feedback

STUDENT DASHBOARD:



Feedback for Class: Organisational Behaviour (HRGM101)

Essay Submitted:

OB, also known as Organisational Behaviour, is the study of how different individuals behave in a group. Its core concepts are Individual Behaviour, Group Dynamics and Organisational Culture. Its key focus areas are Leadership, Communication, Motivation, etc.

Transcript (from Lecture):

Organisational behaviour is the study of how individuals in groups act within organisations and how the behaviour impacts the organisation's effectiveness. It examines individual behaviour, group dynamics, and organisational structure to understand and improve work place dynamics. Ultimately, OB aims to enhance organisational performance, create a positive work environment, and foster better professional relationships. The core concepts are individual behaviour, group dynamics, organisational structure, and the key focus areas are motivation, communication, leadership, organisational culture, change management, and conflict resolution.

Engagement Score:

Moderate engagement

Lecturer Feedback:

Okay!!

 [Back to Dashboard](#)

Conclusion:

All in all, EngageTrack offers real-time insights into student engagement and comprehension. It is an AI-powered platform that aims to improve classroom instruction. It gives teachers the ability to make timely, data-driven decisions that increase the effectiveness of their instruction through integrated modules for lecture transcription, engagement tracking, and essay-based analysis.

Early detection of struggling or disengaged students is made possible by the system, enabling individualised support and intervention. Through individualised dashboards that show feedback, engagement metrics, and lecture content, it empowers students at the same time, promoting self-awareness and active engagement.

EngageTrack's full-stack architecture and user-friendly interface make it suitable for both traditional and virtual classrooms. It promotes a more inclusive, engaging, and successful educational experience by bridging the gap between teaching and learning.

Future enhancements:

- **Set up a student assistant driven by AI:** To encourage self-directed learning, add a conversational agent that can contextually respond to student enquiries based on lecture transcripts.
- **Support transcription and multilingual user interfaces:** To accommodate varied classrooms, enable multilingual support in the transcription module and user interface.
- **Provide notifications and alerts in real time:** Teachers may respond more quickly if they are informed when students showcase signs of inactivity or disengagement.
- **Add gamification elements:** To encourage students and promote active participation, implement badges and progress tracking.

Acknowledgements:

We are extremely thankful to Intel Unnati for giving us the chance to participate in this industrial training as well as for providing us with the necessary tools, guidance, and hands-on learning environment to allow us to work on this creative project.

We are deeply grateful to Mr. Suraj Aravind Bollapragada, our faculty mentor, for his constant encouragement, support, and technical advice during this project.

We also thank our team, which consists of just the two of us, for their hard work in taking full responsibility for the project, from design and planning to testing and execution. Our combined dedication, problem-solving skills, and teamwork were essential to EngageTrack's successful development.

References:

- **FastAPI Documentation** – <https://fastapi.tiangolo.com/>
- **MongoDB Documentation** – <https://www.mongodb.com/docs/>
- **Web Speech API** – https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
- **JavaScript, HTML, CSS**