Department of Computer Science
Faculty of Computing and Information Technology (FCIT) University of the Punjab

AI-Based Cognitive Load Estimation via Natural Activity Monitoring

Final Year Project Proposal

Session 2025-2026

A project submitted in partial fulfilment of the BS in Computer Science
Fall 2025

16 September 2025

Project Registration

Project ID (for office use)						
Nature of project	[ ] **D**evelopment			[✓] **R&D**		
Area of specialisation	Image processing, Computer Vision, Web Development					
Project Group Members						
Sr.#	Reg. #	Student Name	CGPA	Email ID	Phone #	Signature
(i)	Group Leader BCSF22M023	Muhammad Moin U Din	3.5	bcsf22m023@pucit.edu.pk	03028791634	
(ii)	BCSF22M031	Muhammad Junaid Malik	3.41	bcsf22m031@pucit.edu.pk	03265869774	
(iii)	BCSF22M043	Muhammad Subhan Ul Haq	3.25	bcsf22m043@pucit.edu.pk	03338133811	

Plagiarism Free Certificate

This is to certify that, I am <u>Muhammad Moin U Din</u> S/D/o <u>Abdul Hameed</u>, group leader of FYP under registration no <u>BSCSF22M023</u> at Computer Science Department, University of the Punjab, Lahore. I declare that my FYP proposal is checked by my supervisor, and the similarity index is **<u>6%</u>**, that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: <u>16/09/2025</u> Name of Group Leader: <u>Muhammad Moin U Din</u> Signature:

Name of Supervisor: <u>Ms. Tayyaba Tariq</u> Co-Supervisor (if any): <u>N/A</u>

Designation: <u>Assistant Professor</u> Designation: <u>N/A</u>

Signature: ____________________ Signature: <u>N/A</u>

Project Abstract

This project aims at the design and implementation of an artificial intelligence (AI) based **cognitive load estimation system** to monitor the natural user activity to assess the level of mental engagement during study/work sessions. In contrast to the usage of traditional digital wellbeing applications that serve generic usage statistics, this system uses real-time behavioral signals from webcam, keyboard and mouse interaction to determine if a user is overloaded, optimally engaged or disengaged.

Using combinations of **facial expressions**, such as frowning, yawning or raised eyebrows, and eye gaze tracking and blink rate, the system reads signs of fatigue, confusion or distraction. Keyboard typing patterns, frequency of pauses, corrections of errors and behaviors of movement of mouse are also recorded to get more subtle signs of cognitive effort. These features are then run through machine learning models to classify the user's cognitive load into **three** categories: Low, Medium and High.

The solution is intended to give individual feedback as well as aggregated information. For individuals, real-time popups suggest taking breaks in case of **cognitive overload** or increasing focus on the periods of disengagement. A desktop application will be constantly pulling activities in the background, whereas a dashboard will give detailed logs, visualizations and engagement metrics. This project is a practical meeting of the fields of computer vision, machine learning and human computer interaction that contributes to the goal of smarter productivity tools and more personalized learning environments.

Introduction

The project, AI-Based Cognitive Load Estimation via Natural Activity Monitoring, is aimed at creating a system that detects whether users are mentally **overloaded**, engaged or disengaged during digital activities. Traditional digital wellbeing tools only measure screen time, but this project aims at real-time behavioral monitoring for a better and more accurate **assessment** of mental effort.

The system combines several different modalities: computer vision to analyze the facial cues, gaze and blinks from the webcam; keyboard and mouse interaction patterns for pauses, corrections and hesitation; and for collaborative situations. Extracted features will be fed to machine learning models like Random Forests for simpler classification or CNN+LSTM model architectures for temporal signals.

Some of the key sub-tasks are activity monitoring, feature extraction, cognitive load classification and feedback provision through desktop app and web dashboard. The success of the project will be tested with an accuracy test on known data sets such as DAiSEE and CLT as well as a demonstration of **real-time feedback and visualization**. Evidence of achievement will include experimental results, system performance control and a working prototype that tracks and reports user load well.

1. Success Criterion

The project will be considered successful if it can:

- Correctly categorize user cognitive load as Low, Medium, and High states with satisfactory performance in benchmark data sets such as DAiSEE and CLT.
- Providing real-time monitoring using the desktop application without significant delay and system crash, can incorporate multiple sources of input (webcam, keyboard, mouse) and generate consistent outputs.
- Effectively display results on a web-based dashboard which includes logs, visualizations and feedback notification.
- To be accepted by the supervisor as achieving the objectives defined - Possible for practical applications in academic or productivity situations.

2. Related work

Early work by **Sweller** [1] introduced Cognitive Load Theory, which explained that the mental effort we use while solving problems directly affects how well we learn. But this was only a theory and did not suggest any automatic way to measure load.

Later studies focused on practical ways to measure cognitive load. **Anders** et al. [2] collected data in real-world, everyday settings, but they did not build prediction models on top of it. **Bhatti** et al. [3] created CLARE, a dataset for real-time cognitive load estimation, but it was limited in scale and harder to use outside research. **Strle** et al. [4] used physiological signals to estimate load in cars, but their approach was tied only to driving. **Liu et** al. [5] achieved strong results using Multiview learning on multimodal signals, but their models mainly depended on clean lab data that does not work as well in noisy, real-life conditions.

2.1 Research Gaps

- Most prior studies are dataset-driven rather than building end-to-end usable systems.
- Many focus on physiological sensors (EEG, ECG), which are intrusive and impractical for everyday use.
- Approaches are often restricted to controlled labs or domain-specific contexts like driving simulators.

2.2 Our Contribution

This project addresses those gaps by using natural activity monitoring (webcam, keyboard, and mouse inputs) that occur during normal digital work and study sessions. Unlike prior works, our system:

- Automatically detects load in real time using multimodal features.
- Provides actionable feedback (e.g., break reminders) instead of only labeling data.
- Offers practical tools: a desktop application for activity capture and a web dashboard for visualization, making the solution accessible to everyone.

3. Project Rationale

The purpose of this project is to design a system that can estimate the cognitive load of a user by natural monitoring of their activity. Unlike existing digital wellbeing tools that focus only on determining the time of use, this project is focused on real-time mental effort detection using webcam, keyboard and mouse interaction.

a) Motivation

- Growing digital learning and remote work environments require new and improved methods of engagement monitoring and preventing mental fatigue.
- Teachers and supervisors don't have objective indicators of student/worker overload or disengagement.
- Current research systems are either restricted to the laboratory or unimodal (using only facial or physiological signals).

b) Relevance

- Provides real-time feedback for learners and workers that helps them improve the productivity and wellbeing of their activities.
- Serves to engage in the class without being intrusive to the students.
- Contribution to AI and Human-Computer Interaction research through the fusion of multimodal signals.

c) What We Aim to Learn

- What facial cues, gaze patterns, and typing / mouse behaviors can be incorporated into a reliable model of cognitive load.
- Which approach of machine learning model (Random Forest, CNN+LSTM and multimodal fusion) is best for natural activity monitoring.
- How to turn research information into a working system (desktop application + web dashboard).

3.1 Aims and Objectives

3.1.1 Aim

The goal of this project is to design and develop an artificial intelligence (AI)-based system to estimate cognitive load in real-time based on natural activity monitoring (facial expressions, gaze, keyboard, and mouse interactions) to provide meaningful feedback to enhance productivity and engagement in digital learning and work environments.

3.1.2 Objectives

- To collect and preprocess multimodal data (facial cues, gaze patterns, typing and mouse activity) for cognitive load estimation.
- To identify salient features from visual and behavior signals that correlate to low, mid and high cognitive load.
- To build and evaluate machine learning models (Random Forest, CNN + LSTM, multimodal fusion) for cognitive load classification.
- To develop a desktop application for real-time activity capture and monitoring.
- To establish a web-based dashboard for presenting results, trends, and feedback to users and educators.
- To use benchmark data sets and real-time trials to test and validate the system for correctness and usability.

3.2 Scope of the Project

The scope of this project is to design and implement the cognitive load estimation system based on natural activity monitoring that is real-time. The project will use a combination of facial behavior, gaze-tracking and keyboard/mouse interaction capabilities to provide a classification of low, medium and high cognitive load in users and feedback through a desktop application and web-based dashboard.

3.2.1 Project Goals

- To construct a system for estimating cognitive load by combining multiple modalities of signals (visual + behavioral signals).
- To implement machine learning models that can perform classification in real time.
- To provide feedback and visual reports through user-friendly interfaces.

3.2.2 Deliverables

- A desktop application (if required) for the capture of activities (webcam, keyboard, mouse).
- A web dashboard for results, log, and trends presentation.
- A trained AI model capable of classifying cognitive load levels.
- Project documentation, including reports and user manuals.

3.2.3 Features and Functions

- Real-time user activity monitoring.
- Classification of Cognitive Load from Low, Medium and High.
- Feedback notifications (e.g. "Take a break" when there is overload).
- Logging and visualizing engagement trends in the dashboard.

3.2.4 Constraints

- Deadlines for the FYP must coincide with the official calendar of the FYP.
- Deep learning is not guaranteed to succeed, as the models used can be very large and require a significant number of computational resources (GPU/processing power) to train.
- Data availability will depend on the choice of public datasets as well as small-scale pilot data collection.

4. Proposed Methodology and Architecture

The methodology uses a step-by-step pipeline of combining data acquisition, feature extraction, model development and system deployment. The process is intended to be accurate, implemented in real time, and must be practical to use.

4.1 Data Acquisition

- Facial expressions, gaze and blinks from webcam input.
- Keyboard and mouse logging of typing speed pauses and erratic movement.
- Public datasets (e.g. DAiSEE, CLT) and small pilot data will be used.

4.2 Data Preprocessing

- Different video frames are extracted and resized for faster processing.
- Cleaning the facial and gaze signals by MediaPipe/OpenCV to reduce the noise and to smooth the signals.
- Normalize the data of the keyboard and mouse activity so that it can be compared to each other.

4.3 Feature Extraction

- Detect facial cues such as micro-expressions, head movements, and eye blinks.
- Track gaze patterns and determine if user is focused on screen or distracted.
- Capture behavioral signals such as typing mistakes or hesitation of mouse.

4.4 Model Development

- Start with simple models like Random Forest or SVM as baselines.
- Go for advanced deep learning models (CNN + LSTM) for handling time-based patterns.
- Combining the fusion of both the visual and behavioral features in a multimodal fusion for a higher accuracy.

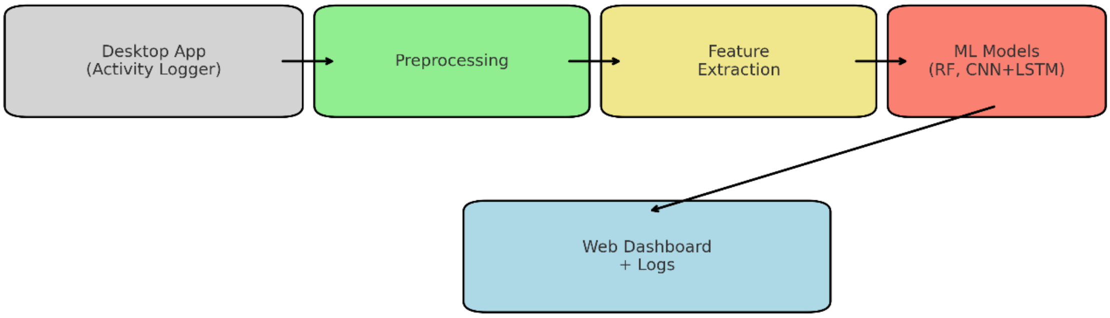
4.5 System Integration

- Build a desktop app that runs in the background and collects real-time activity.
- Set up a backend server to handle preprocessing and to run the trained models.
- Develop a web dashboard for displaying logs, engagement levels, and graphs of trends.

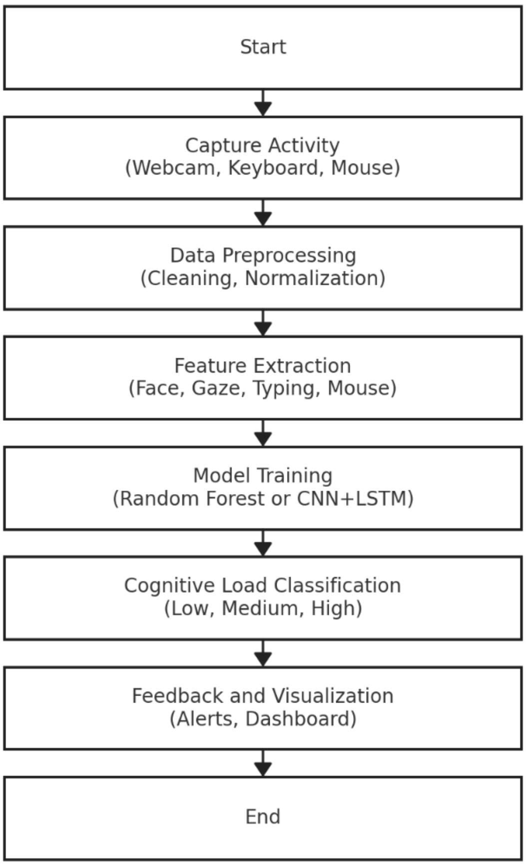
4.6 Feedback Mechanism

- Provide Real-time alerts "You seem overloaded, take a break".
- Teacher's dashboard should offer teachers a view of the class, including trends in overall engagement and cognitive load.

High-Level Architecture



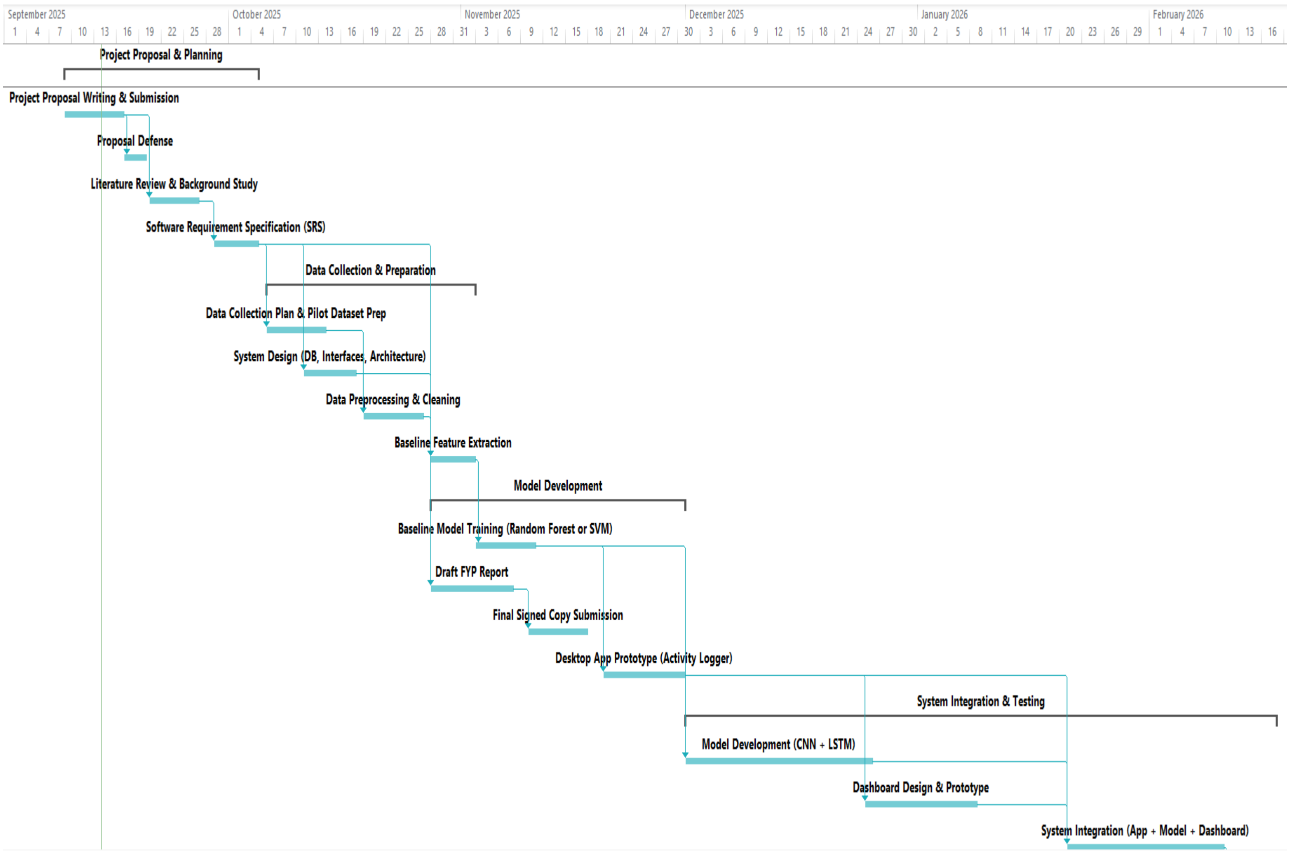
Methodology Flowchart

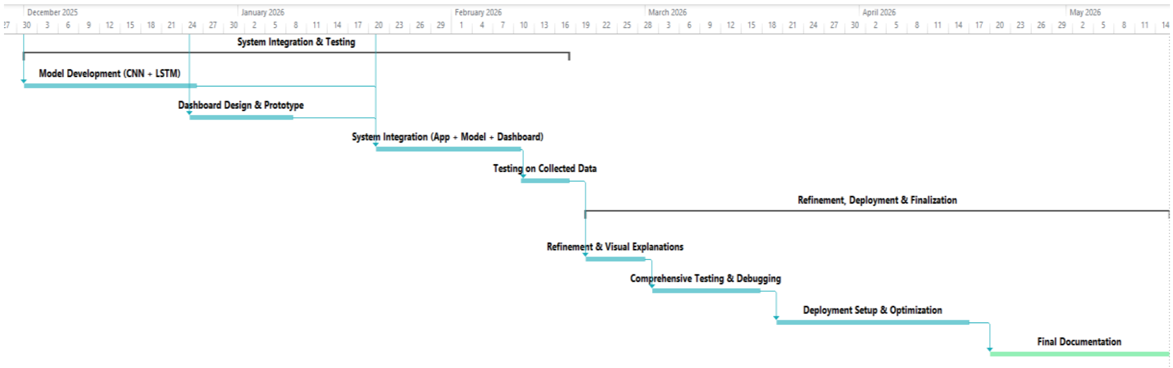


5. Individual Tasks

Team Member	Activity	Tentative Date
Moin, Junaid, Subhan	Project Proposal Writing & Submission	Sep 16, 2025
Moin, Junaid, Subhan	Proposal Defense	Sep 19, 2025
Moin, Junaid	Literature Review & Background Study	Sep 26, 2025
Moin, Subhan	Software Requirement Specification (SRS)	Oct 4, 2025
Moin	Data Collection Plan & Pilot Dataset Prep	Oct 13, 2025
Junaid	System Design (DB, Interfaces, Architecture)	Oct 17, 2025
Moin, Subhan	Data Preprocessing & Cleaning	Oct 26, 2025
Junaid	Baseline Feature Extraction	Nov 2, 2025
Moin, Junaid	Baseline Model Training (Random Forest or SVM)	Nov 10, 2025
Subhan	Draft FYP Report	Nov 7, 2025
Moin, Subhan	Final Signed Copy Submission	Nov 17, 2025
Junaid, Subhan	Desktop App Prototype (Activity Logger)	Nov 30, 2025
Moin, Junaid	Model Development (CNN + LSTM)	Dec 25, 2025
Subhan	Dashboard Design & Prototype	Jan 8, 2026
Moin, Junaid, Subhan	System Integration (App + Model + Dashboard)	Feb 10, 2026
Moin, Subhan	Testing on Collected Data	Feb 17, 2026
Junaid	Refinement & Visual Explanations	Feb 28, 2026
Moin, Junaid	Comprehensive Testing & Debugging	Mar 17, 2026
Subhan	Deployment Setup & Optimization	Apr 16, 2026
Moin, Junaid, Subhan	Final Documentation	May 15, 2026

6. Gantt Chart





7. Tools and Technologies

To design and implement the proposed system successfully, the following tools and technologies will be used:

7.1 Programming Languages & Frameworks

- **Python** - for machine learning model development and preprocessing.
- **JavaScript (React / Next.js)** - for developing the web dashboard.
- **C++/Rust with Tauri/Electron** - for building the cross-platform desktop application (if required).

7.2 Machine Learning & AI Libraries

- **TensorFlow / PyTorch** - for training and deploying deep learning models (CNN, LSTM).
- **scikit-learn** – for classical ML algorithms (Random Forest, SVM).
- **MediaPipe / OpenCV** - for facial feature detection, gaze tracking, and image preprocessing.
- **DeepFace** - for emotion and micro-expression recognition.

7.3 Data Handling & Storage

- **PostgreSQL / MySQL** - for storing logs and processed data.
- **Pandas & NumPy** - for dataset management and feature engineering.

7.4 Development & Collaboration Tools

- **Visual Studio Code / PyCharm** - IDEs for coding.
- **Git & GitHub/GitLab** - for version control and team collaboration.
- **Jupyter Notebook** - for experimentation and model evaluation.

7.5 Deployment & Testing

- **Docker** - for containerizing applications and ensuring reproducibility.
- **Mailpit/Resend (optional)** - for notifications and feedback alerts.
- **Unit Testing Frameworks (PyTest, Jest)** - for validating code functionality.

7.6 Visualization Tools

- **Matplotlib / Seaborn** - for plotting feature trends and model results.
- **Web Dashboard (React Charts / Recharts)** - for real-time visualizations of engagement and cognitive load levels.

8. References

[1] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1202_4

[2] C. Anders, S. Moontaha, S. Real, et al., "Unobtrusive measurement of cognitive load and physiological signals in uncontrolled environments," *Scientific Data*, vol. 11, no. 2, 2024. [Online]. Available: https://www.nature.com/articles/s41597-024-03738-7

[3] A. Bhatti, P. Angkan, B. Behinaein, Z. Mahmud, D. Rodenburg, H. Braund, P. J. McLellan, A. Ruberto, G. Harrison, D. Wilson, A. Szulewski, D. Howes, A. Etemad, and P. Hungler, "CLARE: Cognitive Load Assessment in Real-time with Multimodal Data," arXiv:2404.17098 [cs.HC], 2024. [Online]. Available: https://arxiv.org/abs/2404.17098

[4] G. Strle, A. Košir, J. Sodnik, and K. Stojmenova, "Physiological Signals as Predictors of Cognitive Load Induced by the Type of Automotive Head-Up Display," *IEEE Access*, 2023. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3305383

[5] Y. Liu, Y. Yu, H. Tao, Z. Ye, S. Wang, H. Li, D. Hu, Z. Zhou, and L.-L. Zeng, "Cognitive Load Prediction from Multimodal Physiological Signals using Multiview Learning," *IEEE Journal of Biomedical and Health Informatics*, 2023. [Online]. Available: https://doi.org/10.1109/JBHI.2023.3346205

AI based Cognitive Load Estimation

ORIGINALITY REPORT

6%	5%	0%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	4%
2	www.coursehero.com Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		