ZenMode

A Context-Aware Focus Drift Detection Assistant

Project Proposal

Project Group 2 | CSE299 – Section 4 | Spring 2026 | North South University

Name	NSU ID	Email
Aminul Islam	2321169042	aminul.islam.232@northsouth.edu
Md Shahriar Rakib Rabbi	2321937642	shahriar.rabbi.232@northsouth.edu
Tirtho Mojumdar	2312536042	tirtho.mojumdar.2312536@northsouth.edu

Faculty Advisor: Dr. Mohammad Shifat-E-Rabbi (MSRb)

1 Abstract

Maintaining sustained focus during study or work sessions is increasingly challenging due to constant digital distractions. Traditional productivity tools either rely on rigid blocking mechanisms or simple timers that lack awareness of actual work context. **ZenMode** addresses this gap by offering an intelligent, non-intrusive desktop application that combines real-time activity monitoring with AI-powered context analysis. The system detects when users drift from their stated goals and provides gentle, adaptive nudges to restore focus, respecting user autonomy while fostering mindful productivity.

2 Problem Statement

2.1 Background

The modern digital environment presents an unprecedented challenge to human attention. Students preparing for examinations and professionals completing work assignments face constant distractions from applications, notifications, and content competing for attention. Research demonstrates that after a distraction, it takes approximately 23 minutes to fully regain focus [1], making even brief attention drifts significantly costly to productivity.

2.2 Current Challenges

- **Lack of Self-Awareness:** Users often fail to recognize when their attention has drifted until substantial time has passed on unproductive activities.
- **Rigid Existing Tools:** Most productivity applications rely on website blockers or strict timers that frustrate users who may legitimately need access to certain resources.
- **Context Insensitivity:** Existing solutions do not understand the relationship between user goals and current activities. A programming tutorial may be productive for one task but distracting for another.

2.3 Problem Definition

How can we design a focus assistance system that intelligently detects attention drift based on user-defined goals and current activities, while providing non-intrusive guidance that respects user autonomy and promotes self-awareness?

3 Proposed Solution

3.1 Solution Overview

ZenMode is a **desktop application** designed to help users maintain attention during focus sessions by detecting moments of attention drift and guiding them back to their stated goals. The name reflects the core philosophy: entering an intentional focus state where the system assists users in remaining mentally present.

3.2 Core Features

Feature	Description
Session Management	Users initiate focus sessions by declaring their goal (e.g., "Preparing for CSE115 exam"), which becomes the contextual anchor for activity analysis.
Activity Monitoring	Continuously tracks active windows, browser tabs, and time spent on each activity using lightweight background processes.
AI-Powered Analysis	Uses Gemini API [2] to analyze whether current activities are relevant to the declared goal.
Intelligent Prompting	When potentially distracting activity persists beyond a threshold, the system prompts users to confirm whether the activity is work-related.
Productivity Analytics	Dashboards display productive vs. distracted time, activity breakdowns, and historical trends.

3.3 Key Innovation

The distinguishing feature is **context-aware analysis**. Unlike traditional blockers that categorize websites as universally "productive" or "distracting," ZenMode understands that a YouTube video on recursion is productive for someone studying data structures but distracting for someone writing a history essay.

3.4 User Workflow

1. **Start Session:** User enters a goal description.
2. **Focus Period:** System monitors activities in the background without interruption.
3. **Drift Detection:** If the user spends significant time on a potentially irrelevant activity, a gentle prompt appears.
4. **User Response:** "Yes" marks activity as productive; "No" logs time as distracted with a motivational reminder.
5. **Session End:** Comprehensive statistics help users understand their focus patterns.

4 METHODOLOGY

4.1 TECHNOLOGY STACK

Component	Technology	Justification
Desktop Framework	Electron [3]	Cross-platform support using web technologies
Frontend UI	React 18 [4] + Vite	Modern, component-based architecture
Backend Runtime	Node.js	JavaScript throughout; extensive npm packages; asynchronous operation support
UI Styling	Tailwind CSS + shadcn/ui	Professional appearance; rapid prototyping
Activity Monitoring	`active-win` npm package	Cross-platform window detection
Local Database	SQLite	Local-first storage; zero configuration
AI Analysis	Gemini API [2] (Google)	Intelligent context understanding
Visualization	Recharts	React-native responsive charts

4.2 SYSTEM ARCHITECTURE

The application follows a two-process Electron [3] architecture:

- **Main Process (Node.js):** Handles activity monitoring, AI analysis, database operations, and IPC communication.
- **Renderer Process (React):** Provides user interface including session controls, dashboard, and settings.

4.3 DATA FLOW

1. Background service polls active window information at 5-second intervals.
2. Window titles and session goals are sent to AI API for relevance scoring.
3. If activity is deemed distracting for threshold duration, user prompts are triggered.
4. User confirmations update the database and refine future analysis through caching.
5. UI receives live updates via IPC for real-time dashboard refresh.

5 DEVELOPMENT TIMELINE

Phase	Week	Activities	Deliverables
Foundation	1	Project setup; Electron + React configuration; activity monitoring; SQLite schema	Working app tracking active windows
Intelligence	2	AI API integration; prompt engineering; distraction detection logic	System identifying potential distractions
Analytics	3	Statistics engine; Recharts integration; dashboard creation	Analytics dashboard with visualizations
Polish	4	UI refinement; testing; bug fixes; documentation; demo video	Production-ready application

6 Expected Outcomes

Upon successful completion, ZenMode will:

- Accurately monitor active windows and track session time
- Correctly identify 80%+ of distractions vs. productive activities using AI
- Provide a clean, intuitive UI requiring no technical knowledge
- Operate with minimal system resource usage (<5% CPU, <100MB RAM)
- Deliver actionable productivity statistics and insights

Learning Outcomes

- Cross-platform desktop development with Electron
- Real-time system monitoring and event-driven architecture
- Third-party AI API integration
- Database design and modern frontend development

7 Conclusion

The **ZenMode** project addresses the widespread challenge of attention management in digital environments. By combining proven technologies with modern AI capabilities, we aim to deliver a functional application that helps users maintain focus through intelligent, context-aware assistance rather than rigid restrictions. This project aligns with CSE299 objectives by solving a real-world problem, demonstrating full-stack development capabilities, and producing a tangible, usable product.

References

[1] D. G. U. K. Gloria Mark, "The cost of interrupted work: more speed and stress," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy, 2008.

[2] "Gemini API Docs," Google AI, 2026. [Online]. Available: https://ai.google.dev/gemini-api.

[3] "Electron Docs," OpenJS Foundation, 2026. [Online]. Available: https://www.electronjs.org/docs.

[4] "React," Meta Open Source, 2026. [Online]. Available: https://react.dev/.