

PROJECT ABSTRACT

Pomodoro Focus Timer Desktop App using Electron

A cross-platform Pomodoro focus timer built with Electron: configurable 25/5 focus/break cycles, a task manager with per-task pomodoro tracking, system notifications, an always-on-top mini timer, and a local statistics dashboard — built-to-order with code, installers, report, PPT and viva kit.

01. Title

Pomodoro Focus Timer Desktop App using Electron

02. Introduction / Background

The Pomodoro Technique structures work into focused 25-minute sessions separated by short breaks, with a longer break every fourth session — a simple contract that counters the fragmented attention of modern study and desk work. A browser-tab timer is too easy to lose among distractions; a desktop application with native system notifications, an always-on-top mini timer and a fullscreen focus mode stays present without being noisy. Electron makes such an app practical for a student build: web technologies (HTML, CSS, JavaScript) run inside a desktop shell with Node.js system APIs, and electron-builder packages one codebase into installers for Windows, macOS and Linux. All data in this build stays local — there is no account, no server and no tracking.

03. Problem Statement

Students and developers struggle to sustain single-task focus, and existing timer options are either easy-to-lose browser tabs or bloated productivity suites with cloud accounts. The project addresses this with a focused, offline desktop timer: a Pomodoro state machine (focus → short break → ... → long break) with configurable lengths, a task manager assigning pomodoro estimates to tasks, native OS notifications and sounds at transitions, an always-on-top mini timer and fullscreen focus mode, plus a statistics dashboard aggregating the local session log. The scope is deliberately a focus timer — website blocking and cloud sync are documented as future scope, not included.

04. Objectives

- Implement a Pomodoro timer state machine with configurable focus, short-break and long-break durations and sessions-per-long-break.

- Build a task manager with pomodoro estimates, check-off, reordering and per-task completed-session tallies.
- Integrate native OS notifications and configurable sounds for session transitions.
- Implement an always-on-top mini timer and a fullscreen focus mode.
- Build a statistics dashboard (daily/weekly sessions, focus minutes, completion rate) from a local session log with CSV export.
- Persist tasks and history in a local SQLite database with no network dependency.
- Package cross-platform installers via electron-builder and deliver the full student kit: code, report, PPT and viva Q&A.

05. Proposed Methodology

The app follows Electron's main/renderer architecture: the main process owns the timer state machine, scheduling and OS integration (notifications, always-on-top window, tray), while renderer processes implement the timer, tasks, statistics and settings views in HTML/CSS/JavaScript, communicating over IPC. The state machine transitions focus → short break, inserting a long break every Nth session per settings; each transition fires a native notification, plays the configured sound and appends to the session log. Tasks and history persist in SQLite; the statistics view aggregates the log into daily/weekly charts. electron-builder produces signed-packaged installers for Windows, macOS and Linux from the same codebase. Timer behavior under OS sleep is documented as a limitation rather than engineered around.

06. System Architecture / Working

The system has three parts. (1) Main process: timer state machine, scheduler, notifications, window management (mini timer, fullscreen mode), SQLite access. (2) Renderer views: timer, tasks, statistics, settings, communicating with main over IPC. (3) Packaging: electron-builder installers per OS. The working flow is: add tasks with estimates → start focus session → work until notification → break → repeat to long break → review statistics. All state transitions are logged, making the statistics reproducible from the session history.

07. Hardware / Software Requirements

- Windows 10+, macOS 11+ or a modern Linux distribution (installers included)
- Electron, Node.js, JavaScript, HTML5/CSS3, SQLite, electron-builder
- No internet connection required after installation
- Notification permissions granted at OS level for reminders

08. Expected Outcomes

A polished, installable desktop Pomodoro timer with task tracking, notifications, focus modes and a statistics dashboard, working identically on Windows, macOS and Linux

from one codebase. The reproducible outcome is the working app, its installers and documentation — behavioral claims are limited to implemented features, with OS-dependent behaviors (notifications, sleep) documented as limitations.

09. Applications

- Personal productivity for students during exam and project preparation
- Teaching aid for desktop-app architecture, Electron IPC and state machines
- Template for other Electron utilities (habit trackers, offline tools)
- Portfolio piece demonstrating cross-platform packaging

10. Future Scope

- Website and application blocking during focus sessions
- Optional cloud sync for cross-device session history
- Calendar integration for time-boxed scheduling
- Auto-update support for distributed versions