Startup Program Manager

Project Abstract — projectech.in (built to order) · K-1627

Introduction / Background

Boot time is the most felt performance metric on a PC, and it degrades silently as every installed program adds itself to startup. This project builds a startup program manager: a complete inventory of auto-start entries, per-entry boot-impact scoring, safe enable/disable with undo, delayed starts, a boot timeline visualization, and a simulator that projects boot time as entries are toggled.

Problem Statement

Built-in tools list startup items but give no guidance on what is safe to disable and no picture of how the boot sequence fits together. Users either touch nothing and suffer slow boots, or follow random online advice and break things. A manager that measures, visualizes and protects turns guesswork into engineering.

Objectives

- Inventory auto-start entries across registry, folders, tasks and services.
- Score each entry's boot-time impact from a documented load-profile model.
- Allow safe toggling with risk classification and locked critical entries.
- Visualize the boot sequence as a dependency-aware timeline.
- Simulate projected boot time live as entries are toggled or delayed.

Proposed Methodology

The scanner builds the inventory with publisher, launch point and command line per entry. The scoring engine combines CPU cost, disk cost and dependency depth into an impact score. The timeline renderer lays entries on a time axis honoring dependencies; toggling or delaying an entry updates the projection instantly. Risk rules lock critical entries and gate caution-level changes behind confirmations; all actions append to a reversible history.

System Architecture / Working

Single-file web app: data layer (40+ realistic entries with load profiles), model layer (scanner, scorer, timeline builder, simulator), and UI layer (inventory table, timeline canvas, simulator panel, history). The report documents the mapping from the simulated entry set to real Windows autostart APIs for the native build.

Hardware / Software Requirements

- Hardware: any laptop/desktop with a modern browser (2 GB RAM minimum).
- Software: current Chrome, Edge, Firefox or Safari; no installation.

- Native build (documented): Windows with registry/task-scheduler access.

Expected Outcomes

- A working manager that measurably explains and trims boot time.
- A documented impact-scoring model defensible in the viva.
- A safety-first interaction design: locks, warnings, undo.
- Complete viva kit: report, PPT and Q&A.

Applications

- CS/IT coursework on operating systems and performance.
- IT support staff triaging slow-boot machines.
- Portfolio piece in systems software and data visualization.

Future Scope

- Real-machine measurement via a lightweight agent.
- macOS LaunchAgents and Linux systemd support.
- Crowd-sourced entry reputation database.
- Game/launch profiles that switch entry sets.

This is a built-to-order project. The abstract above summarizes the project as delivered: working implementation, documentation, and viva preparation included.