Command-Captain

by Parker Groneck

Problem

1. Managing terminal commands and device information manually is time-consuming
2. No easy way to store, search, or copy commands and device IPs
3. Lack of visual organization or security for command/device data
4. No simple way to **share commands or device info** with team members

The Plan

- Provides a **centralized GUI** to manage commands and devices
- Supports **searching, filtering, importing, and exporting** data
- Features **copy-on-double-click** for quick command/IP use
- Implements **User Authentication** for secure access
- Adds **visual improvements**: alternating row colors, header borders, and styled treeviews

Sprint 1 Implemented Features

- **Implemented Features:** 5
    1. User Authentication UI
    2. User Authentication backend
    3. Devices Tab UI
    4. Commands Tab UI
    5. Cloud Backend storage

Sprint 1 Implemented Requirements

Completed Requirements: 5

1. Authorization
2. JSON data structure
3. JSON Storage and manipulation
4. Import/Export JSON for devices and commands
5. API or Webbased backend

Sprint 1 Slips and Spills

- Pushing Time
- Didn't Leverage AI

Sprint 1 Wins

- Started early and finished all required code
- Ended up cutting some corrners that payed off in the end.

Numbers

Total Implented Features: 5

Total Implented Requirements: 5

LOC Per Week: 147

Total LOC: 739 lines

Weekly Burndown Rate: 20%

Burndown Rate: 100%

Sprint 2 Goals

Week 1: Major UI Upgrade

Week 2: Expand hashing system for all items

Week 3: Enhance data schema for better tracking

Week 4: Add permission and department locks

Week 5: Clean up and Refactor

Demo