RF-2 - Software Requirements Specification

CTRL ALT Elite

Kevin Ekart / Ethan Gray / Laken Hollen / Huynh Le

October 3, 2025

Individual Contributions

Kevin Ekart

- RF-1 – Social Feasibility
- RF-2 – Specific Requirements
- Documentation assembly and formatting

Ethan Gray

- RF-1 – Economic Feasibility
- RF-1 – Market Research
- RF-2 – System Attributes and Other Requirements

Laken Hollen

- RF-1 – Alternative Solutions
- RF-1 – Project Risks
- RF-2 – Specific Requirements

Huynh Le

- RF-1 – Product Feasibility
- RF-1 – Technical Feasibility
- RF-2 – Introduction

Software Requirements Specification

Objective

We aim to build a simple, affordable, and web based digital task board that helps small teams visually organize their work using a Kanban style layout (“To Do,” “In Progress,” and “Done”). The tool will make it easier for teams to stay aligned, track progress in real time, and collaborate effectively without the complexity or cost of premium SaaS platforms.

Scope

- Create, edit, and delete tasks
- Move tasks between customizable columns (To Do → In Progress → Done)
- Assign tasks to team members
- Add due dates to tasks
- Leave comments on tasks for discussion and updates
- Track a basic history of changes (e.g., who moved or updated a task and when)
- Support multiple projects or boards

Consider applications for future

- Advanced features like AI powered suggestions, time tracking, file attachments, or integrations with third party tools
- Complex user roles or enterprise-level permissions
- Mobile apps (initially responsive web design will ensure it works well on phones and tablets)

Definitions

- **Kanban board**: A visual project management tool with columns (lanes) representing workflow stages. ("To Do," "In Progress," "Done" lanes).
- **Task**: A unit of work assigned to a user, with due date, notes, comments, and change history.
- **Real-time collaboration**: Multiple users can view and edit the board simultaneously with instant updates.

- **MVP (Minimum Viable Product):** A functional version with core features only, built within one semester.
- **WCAG 2.2 AA**: Web accessibility standard ensuring usability for people with disabilities.
- **PostgreSQL**: The database used to store user data, tasks, and logs.
- **Next.js**: The React framework used for the frontend.
- **Flask**: The Python web framework used for the backend.
- **WebSockets**: A technology for real-time, two-way communication between the client and server.

System Overview

The system is a web application that allows users to:

- Create and manage multiple projects.
- Add, edit, and drag-and-drop tasks between lanes.
- Assign tasks, set due dates, add comments, and view activity history.
- Access the board from any device with internet (no installation required).
- Built with Next.js (frontend), Python/Flask (backend), and PostgreSQL (database).

References

- **Slack Pricing**: https://slack.com/pricing
- **Project Collaboration Market Data**: https://6sense.com/tech/project-collaboration
- **Software Developer Salary Data**: https://www.indeed.com/career/software-engineer/salaries
- **WCAG 2.2 Guidelines**: https://www.w3.org/WAI/standards-guidelines/wcag/

Overall Description

Product Perspective

This project is a digital task board that helps teams organize and manage their work together. It works like a virtual whiteboard with lanes (like "To Do," "In Progress," and

"Done"). The system consists of a React/Next.js front end, a Python/Flask API, and a PostgreSQL database. Real-time updates are supported.

System Interfaces

- React with NextJS
    - Provides a fast, reactive, component-based interface
    - Supports real-time task updates and a drag and drop interface for moving tasks between lanes
- Tailwind CSS and ShadCN UI
    - Allow rapid development of polished, responsive user interfaces
    - Offer accessible, prebuilt components for consistency
    - Facilitate rich UI features such as task detail popovers, tooltips, and modal dialogs
- Python with Flask
    - Integrates easily with PostgreSQL
    - Suited for real-time features using WebSockets
    - Supports secure user authentication and session management

User Interfaces

- A responsive web user interface (UI) using React/Next.js, Tailwind CSS, and shadcn/ui component libraries.
- A drag-and-drop feature to move tasks between lanes; task details are displayed in modal form including asignee, due date, notes, comments, and history of changes.

Hardware Interfaces

- Runs on standard clent machines that have a modern browser.
- Deployment on servers using a commodity x86_64 host or cloud VM/container runtime for running Python and PostgreSQL.

Software Interfaces

- React/Next.js front-end, Tailwind CSS, and shadcn/ui.
- Flask API with session functionality and authentication.

- PostgreSQL RDBMS that uses indexed tables for improved performance.

Communication Interfaces

- HTTPS for REST/JSON endpoints.
- Secure WebSocket (WSS) for real-time notifications and board updates.

Memory Constraints

- Client: adequate memory for rendering tasks lists and state of the board from a browser.
- Server: caching of API process, connection pools and work sets of frequently queried data; indexes of a size adequate to handle task and activity log reads.
- Database: sized storage according to expected quantities of projects, tasks, comments, and activity records with nightly backup.

Operations

- Users log in, create projects, add tasks, assign an owner, due date, add comments, and drag tasks from one lane across another.
- Project or admin owners handle the membership settings, permissions, and project settings.
- System conducts regular backups and permits restore drills during operations.

Site Adaptation Requirements

- Environment variables for database credentials, API base URL, and WebSocket endpoint.
- Configurable lanes, task fields, and branding per deployment.
- Accessibility checks (WCAG 2.2 AA) and keyboard-only task movement configured as a CI step.

Product Functions

- Project and task creation, viewing, editing and deletion.
- Drag-and-drop to update status of task.
- Assign members, due date, and add notes/comments.
- See history of activities per task for traceability.
- Real-time collaboration such that several users witness changes right away

User Characteristics

- Team members and leads accustomed to tools like Trello/Asana/Jira, expecting low training overhead.
- Mix of technical and non-technical users; primary needs are clarity, speed, and simple workflows.

Constraints, Assumptions, and Dependencies

- Constraints
    - Must deliver a usable MVP within the course timeline.
    - Performance: large boards may require pagination, incremental loading, and indexes to maintain responsiveness.
    - Accessibility: meet WCAG 2.2 AA checks and provide ARIA labels and keyboard navigation.
- Assumptions
    - Users have reliable internet access and modern browsers.
    - Teams will migrate initial tasks via CSV import or manual entry.
- Dependencies
    - Next.js/React, Tailwind, shadcn/ui for UI.
    - Flask for API and WebSocket support.
    - PostgreSQL for relational storage and indexing.
    - Operational backups and restore procedures; simple caching of common queries for performance.

External Interface Requirements

User Interfaces

- The software shall provide a web-based interface accessible via modern browsers.
- The UI shall include:
  - A login/registration page for authentication.
  - A dashboard listing all projects/boards the user has access to.
  - A board view showing lanes and tasks with drag-and-drop functionality.
  - A task detail popover with fields for assignee, due date, description, comments, and activity history.
  - A settings page for managing user profile and project memberships.
- The interface shall be responsive, supporting desktop, tablet, and mobile displays.

API Interfaces

- The backend shall expose a RESTful API over HTTPS, with endpoints for:
  - /auth/register, /auth/login (user authentication)
  - /boards (create, read, update, delete boards)
  - /tasks (create, read, update, delete tasks)
  - /comments (task comments)
  - /users (user info and permissions)
- All API responses shall be in JSON format.

Database Interfaces

- The backend shall interact with PostgreSQL via an ORM (SQLAlchemy).

Communication Interfaces

- The system shall use WebSockets to broadcast real-time board/task updates to all connected clients.

Hardware Interfaces

- None required beyond client device with internet connectivity.

Functional Requirements

Board Management

- Users shall be able to create, rename, and delete boards.

- Users shall be able to add, rename, and delete lanes within a board.
- Users shall be able to reorder lanes by drag-and-drop.

Task Management

- Users shall be able to create, edit, assign, and delete tasks.
- Users shall be able to move tasks between lanes using drag-and-drop.
- The system shall record timestamps and authors for all task modifications.
- Users shall be able to add, edit, and delete comments on tasks.

User Management

- The system shall allow new users to register with email and password.
- The system shall securely authenticate users with hashed passwords.
- Project managers shall be able to add or remove users from boards.
- Administrators shall be able to manage global user roles.

Collaboration & Real-Time Updates

- The system shall update all users' board views within 1 second of any change.

Activity Logging

- The system shall maintain an activity log of all board and task changes.

Performance Requirements

- Average board load time shall not exceed 2 seconds on a broadband connection
- Updates to tasks/boards shall propagate to all connected clients in under 1 second.
- The system shall handle at least 100 tasks per board without noticeable performance loss.
- Database queries shall execute in under 200ms on average under normal load.

Design Constraints

Standards Compliance

- The system shall use HTTPS for all client-server communication.
- Passwords shall be stored using bcrypt hashing with appropriate salting.
- The database schema shall follow ACID compliance for data integrity.

Logical Database Requirements

Entities and Relationships

- Users
  - Attributes:
    - user_id (PK) – UUID - varchar
    - name - varchar
    - email - varchar
    - password_hash - varchar
    - role – varchar
    - created_at - datetime
    - updated_at - datetime
  - Relationships:
    - Tasks
    - Boards
    - Comments
- Boards
  - Attributes:
    - board_id (PK) – UUID - varchar
    - name - varchar
    - owner_id (FK: Users) - varchar
    - created_at – datetime
    - updated_at - datetime
  - Relationships:
    - Lanes
    - Tasks
    - Activity Logs
- Lanes
  - Attributes:
    - lane_id (PK) – UUID - varchar
    - board_id (FK: Boards) - varchar
    - name - varchar
    - order - int
    - created_at – datetime
    - updated_at - datetime
  - Relationships:
    - Tasks
- Tasks
  - Attributes:
    - task_id (PK) – UUID - varchar
    - lane_id (FK: Lanes) - varchar
    - title - varchar

        - description - text
        - assignee_id (FK: Users) - varchar
        - due_date - datetime
        - status - varchar
        - created_at - datetime
        - updated_at - datetime
    - Relationships:
        - Comments
        - Activity Logs
- Comments
    - Attributes:
        - comment_id (PK) – UUID - varchar
        - task_id (FK: Tasks) - varchar
        - author_id (FK: Users) - varchar
        - content - text
        - created_at – datetime
        - updated_at - datetime
- Activity Log
    - Attributes:
        - log_id (PK) – UUID - varchar
        - user_id (FK: Users) - varchar
        - board_id (FK: Boards) - varchar
        - action - varchar
        - target_type - varchar
        - target_id - varchar
        - created_at – datetime
        - updated_at – datetime
- Users To Boards
    - Attributes:
        - user_board_id (PK) – UUID – varchar
        - user_id (FK: users) – varchar
        - board_id (FK: boards) - varchar

Database Requirements

- Must support referential integrity between tasks, lanes, boards, and users.
- Must allow indexes on frequently queried fields.
- Must allow scaling to support larger boards and multiple projects per user.

Software System Attributes

Reliability

- Robust error handling should be implemented so that errors don't crash the system.
- If a user inputs an invalid query, an appropriate error message should be returned, and the system should be unaffected.
- The system should be tested thoroughly, and any bugs found during testing should be fixed.

Availability

- The system should be available for use at least 99.9% of the time.
- Ideally, the only downtime should come from updates to the code. These downtimes should be minimal, lasting for a minute or two at most.

Security

Resistance to Attacks

- The system should at least be secure enough to resist common attacks such as SQL injection and DDoS attacks. Any successful attack should require significant effort and the use of specialized attack vectors.
- The system should be built in a way so that even if a successful attack occurs, no harm should be done to any individuals using the system.

User Security

- Users are required to enter an ID and password to access the system.
- Users should only have exactly the permissions they need, no more.
- User credentials should be securely stored in a location where they are unavailable to non-administrators.

Maintainability

- All notable functions and objects in the codebase should be documented, detailing what they are, what do they do, and how to properly use them.

- The code should be easy to read, avoiding “spaghetti code”.
- Test cases should be independent of the code, so that they still work properly even if the code is updated.

Portability

- The app is currently planned to work on the Ubuntu Linux operating system.
- The app is currently planned to use AWS for online connectivity, so devices using the app will have to support AWS.
- No support for other operating systems is planned at this time.

Key Personnel Information

KEVIN EKART

SENIOR REACT DEVELOPER 2026 CARDINAL LANE, JEFFERSONVILLE, IN, 47130, UNITED STATES 812-697-0048

° Details °

2026 Cardinal Lane, Jeffersonville, IN, 47130, United States
812-697-0048
kevin.ekart@gmail.com

° Skills °

PHP (10+ years)

Python (7 years)

MySQL (10+ years)

MongoDB (8 years)

JavaScript (10+ years)

TypeScript (5 years)

NodeJS (4 years)

React (7 years)

React Native (3 years)

AWS (9 years)

Serverless (7 years)

Postgres (5 years)

NextJS (3 years)

Employment History

Senior React Developer at Forecastr, Louisville, KY

May 2022 — Present

- Helped transition existing platform from React to NextJS.
- Helped improve speed of the application by 60%.
- Built integration pipeline using AWS Glue to pull and aggregate data from multiple sources including QuickBooks Online, Xero, and Hubspot.

Owner and Software Engineer at Echelon Kinetic Art & Multimedia, Jeffersonville, IN

May 2012 — Present

- **Adjunct World**
  - Took over existing code base for maintenance and enhancements using PHP, MySQL, HTML, jQuery, JavaScript, and CodeIgniter.
  - Built course sign up and checkout that integrates with Stripe to process payments and keep inventory of available seats.
- **Unnamed Press**
  - Built and designed online store and admin for store using PHP, MySQL, HTML, jQuery, JavaScript, and Laravel.
  - Built cart and checkout system that integrates with Stripe to process payments.
- **Rally78**

    - Built and designed API for mobile application using PHP, Python, NodeJS, MongoDB, and Lumen.
    - Built mobile application using React Native and TypeScript.
    - Built real-time chat interface in mobile app utilizing NodeJS and integration with Pusher.
    - Served as technical lead during conversations with the USTA about partnering.
- **Core Integrated Marketing**
    - Built integration between Evosus and Sharpspring using PHP and Laravel.
    - Built integration between Salesforce and Sharpspring using PHP and Laravel.
    - Built integration that scans SFTP server for new files, parses them, and loads them into Sharpspring using PHP and Laravel.

Lead Data Architect at Capture Higher Ed, Louisville, KY

October 2020 — May 2022

- Took over the data warehouse project started by an outgoing employee.
    - Optimized scripts that export data from the production database into the data warehouse using Python, MySQL, AWS S3, AWS Glue, and AWS Athena.
    - Reduced runtime of existing scripts from 23 hours to 8 hours.
- Designed and built business intelligence reports using Python, React, and AWS Athena.
- Planned next generation of data warehouse to reduce runtime and costs.
- Operated as technical advisor for product management group, data exchange group, and operations group.

Lead Software Architect at Capture Higher Ed, Louisville, KY

May 2017 — October 2020

- Designed architecture for fourth iteration of company platform (Engage 4) moving from a monolithic style architecture to a microservices design.
- Built initial prototype of Engage 4 using PHP, HTML, MySQL, and Laravel.
- Reimagined how the company sends bulk emails to improve and maintain sending reputation.
    - Developed plan for both IP warmup and domain warmup.
    - Developed plan for avoiding spam traps.
    - Improved open rates from 12% to 28%
    - Improved click through rates from 3% to 12%.
- Operated as technical advisor within the product management group.
- Operated as top tier support for development group to help troubleshoot and resolve issues with the platform.

Manager of Product Development at Capture Higher Ed, Louisville, KY

July 2015 — May 2017

- Oversaw and managed team of 5 developers.
- Operated as a working manager spending 60% of my time coding and 40% managing the team.
- Instituted 2-week sprints using Jira to track progress.
- Designed, built, and oversaw the implementation of the third iteration of the company's platform (Engage 3) using PHP, HTML, jQuery, JavaScript, MySQL, and Laravel.
    - Built email service capable of queueing and sending millions of emails daily via Mailgun.
    - Built tool that would create screenshots of emails to share with clients for approval using NodeJS.

- Moved platform from a single instance for all clients to an instance per client for scaling.

Software Engineer at Capture Higher Ed, Louisville, KY

April 2013 — July 2015

- Designed and built the second iteration of the company's platform (Engage 2) using PHP, HTML, jQuery, JavaScript, MySQL, and Laravel.
- Moved the platform out of single PHP file to an MVC framework using a monolithic design.
- Improved efficiency of file imports reducing import times by up to 400%.
- Built mobile version of Engage 2 using Apache Cordova.
- Built prototype of company's flagship product for tracking and identifying leads on clients' sites using JavaScript, PHP, and Slim.

Software Engineer at The Learning House, Louisville, KY

April 2009 — April 2013

- Worked to design and build company's internal ERP system (Grail) to manage clients and workloads.
- Worked to design and build company's partner portal that allow clients to view and export reports.
- Built integrations from company's ERP and partner portal to clients' Moodle LMS platforms to gather course data and launch courses.
- Started building media library platform for managing images, videos, and other types of content for publication.
- Built real-time chat module with rooms for each course for Moodle to replace existing, buggy chat module using PHP, HTML, jQuery, and NodeJS.

Computer Hardware Specialist at Spalding University, Louisville, KY

March 2008 — April 2009

- Worked as primary contact for resolution to technical issues on campus.
- Responsible for installing, upgrading, and maintaining computer hardware and software for entire campus.
- Managed server network for campus library.
- Managed computer and software inventory.

Education

Bachelor of Science in Computer Science, Indiana University Southeast, New Albany, IN

January 2022 — Present

Undergraduate Certificate in Cybersecurity, Indiana University Southeast, New Albany, IN

January 2022 — May 2025

Associates in Multimedia, ITT Technical Institute, Louisville, KY

March 2006 — August 2008

Awards and Recognition

Forecastr – Speak the Truth Core Values Award

July 2023

Capture Higher Ed - MVP

August 2015

Business First - 20 People to Know - Technology & Innovation

November 2015

Capture Higher Ed - MVP

August 2013

Ethan Gray

1237 N Highway 31, Austin, Indiana (812) 216-0436 egray1333@gmail.com

EDUCATION

Austin High School Austin, Indiana

Core 40 and Academic Honors May 2022

GPA: 3.604

Ivy Tech Community College Sellersburg, Indiana

Associate of General Studies in General Studies May 2022

Indiana University Southeast New Albany, Indiana

Bachelor of Science in Computer Science Expected: May 2026

GPA: 3.329

EXPERIENCE

Math Tutor, Austin High School January – May 2020

- Greeted customers and treated them with kindness and respect, making them feel welcome in the area
- Worked with other math tutors to aid students in completing various math problems
- Maintained a clean learning environment

Dishwasher, Cracker Barrel August 2025 – Current

- Worked with other dishwashers to clean dishes and stock them in a timely manner
- Cleaned the dish room when dirty and at the end of the day
- Assisted in other areas as needed, such as taking the trash out

LAKEN HOLLEN

8487 S Riddle Rd | Leavenworth, IN 47137

812-968-5485 | Lakendawn0813@gmail.com

OBJECTIVE

Aspiring IT professional pursuing a B.S. in Computer Science with hands-on experience in tech support, digital tools, and community-focused projects.

EDUCATION

Indiana University Southeast, New Albany, IN

Bachelor of Science in Computer Science (Expected Graduation: May 2026)

- Minor: Science/Mathematics
- Dean's List: Every semester attended

Relevant Coursework:

- Programming Fundamentals
- Foundations of Digital Computing
- Introduction to Computer Software Systems
- Computer Programming II
- Intro to Operating Systems
- Computer Structures
- Computer Security
- Intro to Digital Forensics
- Data Structures

Crawford County High School, Marengo, IN

Academic Honors Diploma (May 2022)

- GPA: 3.8 (Unweighted), 4.0 (Weighted)
- National Honor Society (2020–2021)
- Honor Roll (Fall/Spring 2018–2020)
- Booster Club Member (2018–2021)

WORK HISTORY

Philanthropy/ IT Intern

Community Foundation of Crawford County | Marengo, IN | January 2025 – Present

- Provide IT support for meetings and events, including setup and troubleshooting of devices, Microsoft SharePoint, and Office 365 tools
- Developed the Crawford County Community Resource Guide, a directory of services for low-income residents seeking food, housing, healthcare, and other assistance
- Supporting outreach efforts through nonprofit site visits and survey development to better communicate the missions, needs, and impact of local organizations
- Contribute to daily operations and special projects across departments, demonstrating adaptability and a proactive mindset
- Selected as the facilitator for the Summer 2025 Code IT Academy, a tech training program hosted by CFCC in collaboration with Ivy Tech and The Mill, aimed at expanding digital skills in the community

Cook - Dietary Department

Todd Dickey Nursing and Rehabilitation – Leavenworth, IN | December 2021 – January 2025

- Prepared breakfast and lunch, often simultaneously, for up to 60 residents, ensuring meals met nutritional standards, dietary restrictions, and facility schedules.
- Efficiently organized and executed tray line service for both breakfast and lunch, helping to maintain a timely and accurate meal delivery system.
- Maintained strict sanitation standards and collaborated with kitchen and nursing staff
- Balanced workflow under tight deadlines to ensure timely service

---

PROJECTS

- **Disk Analyzer Tool** – Built a Python script to analyze disk usage and generate reports

- Board Game Website – Designed a fantasy strategy board game site with HTML/CSS and JavaScript, including dynamic forms and gameplay summaries
- Community Resource Guide – Compiled and organized local nonprofit information into a printed guide distributed to residents seeking food, housing, and healthcare assistance.

TECHNICAL SKILLS

- Programming Languages: Python, Java, C, C++, Assembly, JavaScript
- Tools: Microsoft Office, SQL (basic), Git, IntelliJ, PyCharm, Node.js, Visual Studio Code, Virtual Box, ChatGPT, Excel (basic)
- Operating Systems: Windows, Linux
- Data Analysis & Troubleshooting

SOFT SKILLS

- Communication, Time Management, Problem-Solving, Adaptability, Teamwork

HUYNH LE

CONTACT

281 - 966 - 5277
lehuynh228@gmail.com
Louisville, KY 40228

TECHNICAL SKILLS

Networking: TCP/IP, LAN/WAN, Wi-Fi basics, troubleshooting principles

Cybersecurity: Security fundamentals, firewalls, VPNs, cryptography basics

Programming & Scripting: Java, Python (basics), Bash (exposure)

Operating Systems: Windows, Linux (familiarity)

Tools & Platforms: Wireshark, VirtualBox, VMware, Packet Tracer

Analytical & Soft Skills:

- Problem-Solving & Logical Reasoning
- Data Analysis (Packet/Log review - conceptual)
- Attention to Detail
- Communication (Written & Verbal)
- Teamwork & Collaboration
- Adaptability & Eagerness to Learn

EDUCATION

Indiana University - Southeast

Expected Graduation - May 2026
B.S. Computer Science – Cyber Security

Houston Community College

2017 – 2019
A.S. Computer Science

Industrial University Of Ho Chi Minh City

2009-2013
B.A . Business Administration

Certification In Process

CompTIA Security+
(Expected Completion: December 2025)

CAREER OBJECTIVE

Motivated Computer Science student with a strong foundation in networking, cybersecurity, and technical support. Seeking an Entry-Level Computer Support Assistant position at the Help Desk to leverage my troubleshooting skills, customer service experience, and passion for technology to resolve user issues efficiently and contribute to a collaborative IT team.

PROJECTS

Virtualized Network Setup with Wireshark Traffic Analysis

Description: Configured a virtualized network using VMware/VirtualBox with multiple subnets and analyzed traffic using Wireshark to identify protocol (HTTP, DNS) and troubleshoot connectivity issues.

Skills : Virtualization, subnetting, Wireshark, packet analysis, troubleshooting.

Simulated Network Attack Detection Using Wireshark

Description: Simulated a basic ARP spoofing attack in a virtual environment and used Wireshark to detect malicious traffic, documenting findings and mitigation steps.

Skills: Wireshark, cybersecurity, virtualization, network monitoring.

File Recovery and Analysis

Description: Used tools like Autopsy to recover deleted files from a disk image and analyzed file details (metadata) to track user activity.

Skills Demonstrated: File recovery, metadata analysis, digital forensics.

Hidden Data Detection

Description: Used tools like Steghide to find and extract hidden data in image files. Documented the steps and results.

Skills: Steganography, data concealment detection, digital forensics.

Honeypot with Fake Billing Page

Description: Developed a honeypot system with a fake billing page using Python, HTML, and CSS to collect attacker credentials (ID, username, password) and log malicious activity.

Skills: Python programming, web development (HTML/CSS), cybersecurity, ethical hacking, and threat detection.

Password Strength Checker

Description: Developed a Python program to evaluate the strength of passwords based on criteria like length, use of uppercase/lowercase letters, numbers, special characters, and resistance to common patterns.

Skills: Python programming, cybersecurity, password security, algorithm design.

WORK EXPERIENCE

ASSISTANT MANAGER *Anthony Vince Nail Spa* *2020 - Present*

- Led and trained a team of 13+ employees, fostering teamwork and ensuring excellent customer service.
- Resolved client concerns to improve satisfaction and reduce complaints by 20%.
- Organized employee schedules, optimizing for peak times and seasonal demand.
- Maintained staff records (salaries, PTO, schedules) using Microsoft Office Suite.
- Managed inventory supplies using Excel to track stock levels.
- Provided day-to-day customer support and bilingual translation services (English/Vietnamese)