PROJECT PLAN

Prototype for Production Scheduling Automation

Murtaza Ali Khokhar

[*murtaza.khokhar@ebm.com.pk*](mailto:murtaza.khokhar@ebm.com.pk)

# Project Overview

This project aims to develop a prototype system to automate production scheduling in a factory setting. The primary goal is to efficiently distribute the production of various products across different production lines throughout the month to meet target outputs. The system will ensure optimal use of manufacturing resources while preventing machine overload and reducing downtime. By automating production scheduling, the factory can enhance productivity, maintain consistent product quality, and minimize operational costs.

# Project Objectives

## Primary Objective

## Develop a system to automatically create and manage production schedules based on monthly product targets.

## Secondary Objective

* Distribute production tasks across multiple manufacturing lines to prevent machine overload and ensure even workload distribution.

# Scope of Work

Requirement Analysis & Planning

* Document the specific requirements for the automated scheduling system, including the types of products, number of production lines, machine capabilities, and operational constraints.

System Design and Prototype Development

* Design a high-level architecture for the automated scheduling system, including data flow diagrams, communication protocols, and system components.

Development

* Develop a simple, user-friendly interface to display production schedules and allow for manual adjustments if necessary.

Deployment and Testing

* Deploy the prototype in a selected section of the factory to test its performance in a real-world environment.

# Project Timeline

## Total Duration: 2 Months (8 weeks)

|  |  |
| --- | --- |
| Week 1 | **Project Kickoff**   * Conduct a project kickoff meeting. * Define project goals, objectives, and deliverables.   **Requirements Gathering**   * Engage with production managers and line supervisors to gather requirements. * Document functional and non-functional requirements. * Document user stories * Identify constraints and limitations for the prototype.   **Requirements Review**   * Finalize and get approval on the requirements specification document.   **Basic System Design**   * Draft a simple system architecture (e.g., a flowchart). * Plan basic data structures |
| Week 2 | **Data Model Design**   * Create basic data models using Pandas for handling production data (e.g., machine capacities, production targets). * Set up example datasets in CSV format for initial testing.   **UI/UX Design**   * Sketch out how the user will input data and view scheduling results. |

|  |  |
| --- | --- |
| Week 3 | **Research and Planning**   * Review basic scheduling algorithms suitable (e.g., simple heuristics, linear programming using PuLP). * Plan out the core logic for distributing production tasks.   **Implement Scheduling Algorithm**   * Code the scheduling logic in Python. |
| Week 4 | **Backend Development**   * Develop basic backend functionality to handle data input/output (e.g., loading CSV files, processing data with Pandas). * Integrate the scheduling algorithm with the data model.   **Initial Backend Testing**   * Test backend processes with various datasets to ensure correct scheduling output. |
| Week 5 | **Implement Frontend (CLI/GUI)**   * Develop the CLI or basic GUI using Tkinter.   **Frontend-Backend Integration**   * Connect the frontend with backend modules, enabling full interaction between user inputs and the scheduling system. * Perform basic tests to ensure the system functions end-to-end. |
| Week 6 | **Comprehensive Testing**   * Conduct thorough testing of the entire system (frontend, backend, algorithm). * Use various test cases to check how the system handles different production scenarios. |
| Week 7-8 | **Refinement**   * Address any issues found during testing.   **User Testing**   * Simulate real-world usage by testing the system with larger datasets. * Collect feedback (if possible) from potential users or simulate different user scenarios. |

# Deliverables

* A prototype algorithm that can distribute production tasks across lines and timeframes based on predefined targets and constraints.

# The developed prototype software, including front-end and back-end components, integrated with existing production systems.

# Constraints

* **Limited Scope**: The prototype will focus on automating production scheduling for a specific set of products and a limited number of production lines, rather than the entire factory. This approach will make the project more manageable and allow for targeted testing.
* **Resource Availability**: The system will assume a fixed availability of machines and labor, without accounting for unexpected downtimes or labor shortages due to unforeseen circumstances.
* **Budget and Time Constraints**: The project will operate within a constrained budget and timeline, prioritizing essential features for a functional prototype over additional enhancements or refinements.
* **Real-Time Adjustments**: While the system will allow some level of dynamic adjustment, real-time rescheduling capabilities may be limited in the prototype to maintain simplicity and focus on core functionalities.

# Technology Stack

|  |  |
| --- | --- |
| **Programming Language** | Python |
| **Development Environment** | Jupyter Notebook |
| **Version Control** | GitHub |
|  |  |