

## **Primary Goal**

Generate a time-based schedule per machine showing:

- Which process step for which product batch at what time.

## **Step 1: Extract Required Data from Excel**

From the file Global RampUp HM DCC rev. 22.1.0.xlsx:

### **1. Process Sheet**

- Process Steps (routing sequence)
- Machine Names (e.g., Sigma 688, Alpha 550)
- Cycle Times per step
- Equipment (optional for scheduling)
- Precedence relationships (step order)

### **2. Data Sheet**

- Available time per machine (hours per year, shifts per day)
- Working days and shift structure

### **3. Demand**

- Product IDs and quantities (from Frontpage or Process sheet)

## **Step 2: Create Data Layer**

- Store extracted data in SQLite tables:
  - Machines (machine\_id, name, available\_time)
  - process\_steps (step\_id, product\_id, machine\_id, cycle\_time, precedence)
  - demand (product\_id, quantity)

## **Step 3: Build Scheduling Model**

Use MILP (Mixed Integer Linear Programming) or Constraint Programming:

### **• Decision Variables:**

- start\_time[step, machine]
- end\_time[step, machine]

- **Constraints:**
  - Machine capacity (cannot run two steps at same time)
  - Precedence (step i before step j)
  - Available time per machine
  - Optional: minimize setups/tool changes
- **Objective:**
  - Minimize makespan (total completion time)
  - Or maximize throughput

Libraries:

- Pyomo for optimization
- Pandas + SQLAlchemy for data handling

#### **Step 4: Generate Output**

- **Production Schedule Table:**
  - Rows: Machines
  - Columns: Time slots
  - Fill: Process step + product batch
- **Visualization:**
  - Gantt chart using Plotly or Matplotlib
  - Export as Excel or show in Streamlit

#### **Step 5: Streamlit UI**

- Upload Excel file
- Select optimization goal (e.g., minimize makespan)
- Display:
  - Gantt chart
  - Table view
- Filters: Machine, product, time period

➤ **Hierarchical Database Structure**

**Level 1: Product**

- **product\_id**
- product\_name
- demand\_quantity
- due\_date

**Level 2: Process Routing**

- **process\_step\_id**
- product\_id (Foreign Key - FK)
- step\_order (sequence)
- machine\_id (FK)
- cycle\_time
- setup\_time
- precedence (next step)

**Level 3: Machine**

- **machine\_id**
- machine\_name
- available\_time
- capacity
- tool\_change\_time

**Level 4: Schedule**

- **schedule\_id**
- machine\_id (FK)
- process\_step\_id (FK)
- start\_time
- end\_time

- batch\_id

## Optional Supporting Tables

- Buffer (if later integrated): station\_id, buffer\_size, material\_feed\_time

## Streamlit Output Layout

### 1. Header

- **Title:** *Production Schedule Dashboard*
- **Subtitle:** *Optimized Machine Time Allocation*

### 2. Filters (Sidebar)

- **Select Machine(s):** Dropdown (M1, M2, M3, ...)
- **Select Product Batch:** Dropdown
- **Select Time Range:** Date & Time picker
- **Optimization Goal:** Radio buttons (Minimize Makespan / Maximize Throughput)

### 3. Main Display

#### A. Gantt Chart

- Horizontal bars representing machines.
- X-axis: Time slots (e.g., 08:00, 09:00, ...).
- Each bar segment labeled with:
  - Process step (e.g., Step 4)
  - Product batch ID
- Color-coded by product or process type.

#### Example:

Machine M1: | Step 4 (Batch A) | Step 5 (Batch A) |

Machine M2: | Setup | Step 3 (Batch B) |

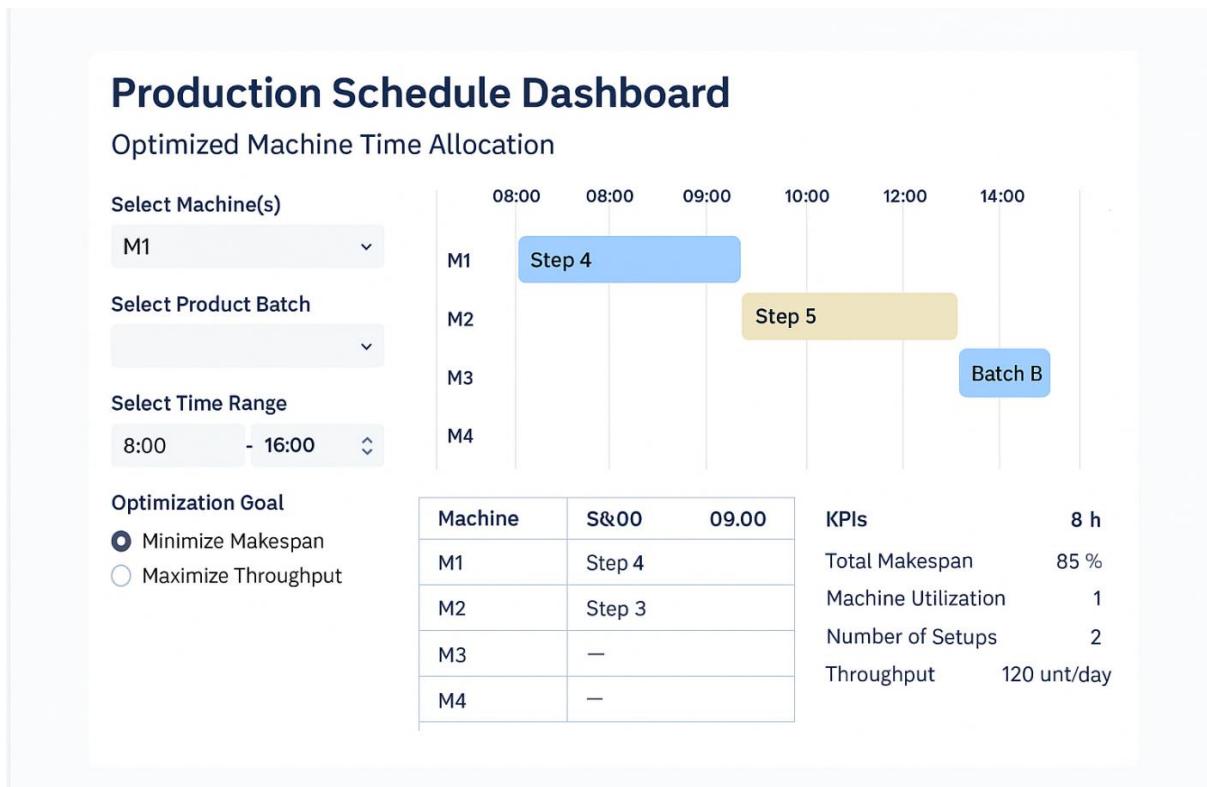
## B. Tabular View

Machine	08:00	09:00	10:00
M1	Step 4	Step 5	Step 6
M2	Setup	Step 3	Step 4

## C. KPIs Section

- Total Makespan:** e.g., 12 hrs
- Machine Utilization:** e.g., M1 = 85%, M2 = 78%
- Number of Setups:** e.g., 5 setups
- Throughput:** e.g., 120 units/day

Expected Streamlit UI for Production Schedule (General visualization)



## **Key Elements in the UI**

### **1. Sidebar Filters:**

- Select Machine(s)
- Select Product Batch
- Select Time Range
- Optimization Goal (Minimize Makespan / Maximize Throughput)

### **2. Main Display:**

- **Gantt Chart:**

- Horizontal bars for machines (M1, M2, M3, M4).
- Time slots on X-axis.
- Colored blocks for process steps and batches.

- **Table View:**

- Machine vs Time slots.

- **KPIs:**

- Total Makespan
- Machine Utilization
- Number of Setups
- Throughput