

ISTE - VESIT
PRESENTS



INVICTUS



24-HOURS
HACKATHON | ONLINE

SPONSORS



Electrolyte Solutions

An ISO 2015:9001 Certified Company

Office Address: Unit No. 11, 3rd floor, B -Wing, Gami Industrial Park, TTC Industrial Area Pawane MIDC, Navi Mumbai – 400710 | **Email:** contact@electrolytesoln.com | **GSTIN:** 27AJYPY7934L1ZS

Title: Inventory Automation & Consumption Analytics for PCB Manufacturing

Hackathon Mode: Online Hackathon | 14–15 February | Duration: 24 Hours

Organizer: Electrolyte Solutions

Context:

Electrolyte Solutions operates a web application used to capture PCB production data and component consumption details. Currently, the system lacks an automated inventory management mechanism, leading to manual stock tracking, delayed procurement decisions, and limited visibility into consumption trends.

Participants are required to design and develop an Inventory Management Module that integrates with PCB consumption data and enables real-time stock control along with procurement alerts.

Objective:

The objective of this hackathon is to build a component-level inventory management system that:

- Automatically deducts component stock based on PCB production
- Tracks component consumption history
- Identifies low-stock components
- Generates procurement triggers when stock falls below defined thresholds

The solution must be achievable within one hackathon day and follow the prescribed technology stack.



Problem Definition:

Each PCB consumes predefined quantities of electronic components.

Example: PCB-A requires 1 unit of a 10 μ F capacitor. If the inventory has 1000 units of the 10 μ F capacitor, producing one PCB-A should automatically reduce the inventory to 999.

The system must ensure:

- Accurate stock deduction
- No negative inventory values
- Clear visibility of component consumption and shortages

Mandatory Technology Stack:

Backend:

- Node.js
- Express.js
- PostgreSQL
- pg (PostgreSQL client)
- JWT (Authentication)
- Multer (File upload)
- xlsx / exceljs (Excel import and export)

Frontend:

- React.js

Database:

- PostgreSQL

Use of alternative backend stacks or databases is not allowed.



INVICTUS



24 - HOURS
HACKATHON | ONLINE

SPONSORS



Functional Requirements (MVP):

1. Component Inventory Management

- Add, view, and update components
- Fields required:
 - Component Name
 - Part Number
 - Current Stock Quantity
 - Monthly Required Quantity

2. PCB-Component Mapping

- Define components used in each PCB
- Specify quantity per component per PCB

3. Automatic Stock Deduction

- Deduct component quantities upon PCB production entry
- Block transactions if available stock is insufficient
- Maintain component consumption history

4. Procurement Trigger Logic

- If current stock is less than 20% of the monthly requirement:
 - Mark the component as Low Stock
 - Generate a procurement trigger record
- Actual purchase order integration is not required



5. Analytics Dashboard

- Component-wise consumption summary
- Top consumed components
- Low-stock component list

6. Excel Import and Export

- Import component inventory using Excel files
- Export inventory and consumption reports to Excel

7. Authentication

- JWT-based authentication
- At least one role: Admin
- Inventory modification APIs must be protected

Explicit non-goals:

The following are not required:

- Advanced forecasting or AI/ML
- Payment systems
- Cloud deployment
- Complex role-based access control
- ERP or third-party system integrations

ISTE - VESIT
PRESENTS



INVICTUS



24 - HOURS
HACKATHON | ONLINE

SPONSORS



Expected Deliverables:

- Working React frontend
- Functional Node.js and Express backend
- PostgreSQL database schema
- Live demo showing:
 - PCB entry and stock deduction
 - Low-stock procurement trigger
 - Analytics dashboard
- README with setup and run instructions

Outcome:

- Top 3 teams will be declared winners
- Winning teams will be considered for internship opportunities at Electrolyte Solutions
- Selected interns may continue development of this module for real production use

Evaluation Focus:

- Correctness of logic over UI appearance
- Functional completeness over feature count
- Data integrity and system reliability over visual polish