

## **FASHION & MERCHANDISING DIGITAL OPERATIONS CHALLENGE**

### **Hackathon Problem Statements — Web Dashboard Edition**

Today's fashion businesses face unprecedented operational challenges: fast-moving inventories, multi-platform order flows, complex product variants, rising customer expectations, and fragmented back-end systems spread across spreadsheets, emails, courier apps, and manual checks. As brands scale, maintaining accuracy, speed, and visibility becomes increasingly difficult without strong digital infrastructure.

Your task is to build a set of advanced web dashboards that help modern fashion and merchandising businesses streamline operations through automation, real-time workflow visibility, and intelligent data handling.

#### **Each dashboard must include:**

##### **User (Seller) Dashboard**

- Customisable settings
- Thresholds, rules & workflow controls
- Notifications & operational alerts
- Visual analytics & status overviews
- Behaviour-based automations (low-stock triggers, order status rules, return logic, etc.)

##### **Admin Dashboard**

- Oversight & monitoring across all sellers
- Risk detection and anomaly flagging
- Compliance & verification workflows
- Insights, trends, and operational analytics
- Control panels for approvals, escalations, and policy enforcement

Each dashboard represents a mission-critical process inside a fashion business, and although teams will build only one, every solution must feel like part of a unified, scalable product ecosystem designed to power the next generation of fashion operations.

---

### **PROBLEM 1: Real-Time Inventory Tracking Dashboard**

For Apparel, Footwear, Accessories, and Seasonal Collections Problem Description

Fashion brands deal with volatile inventory: sizes, colours, SKUs, seasonal merchandise, fast-moving pieces, and slow-moving dead stock. Manual tracking leads to fulfilment errors and broken customer promises.

Your task is to build a web-based Inventory Dashboard that gives sellers real-time visibility and gives admins complete oversight of inventory activity across the platform.

---

## **Seller Dashboard — Required Features**

- Full inventory table for all products, variants, colours, and sizes.
- Add/update stock quantities with reason codes (restock, damaged, returned, QC failed).
- Tag inventory by season (SS25, FW25, festival drops).
- Bulk stock upload via CSV with mapped fields.
- Alerts for low stock thresholds, upcoming expiry (for material-based products), or fast-moving items.
- Timeline of stock movement per SKU (e.g., “+20 added by Seller A at 3:45 PM”).
- Visual stock indicators (green: healthy, yellow: low, red: critical).

## **Admin Dashboard — Required Features**

- Oversee all inventory actions from every seller.
- Approve/flag suspicious updates (e.g., stock jumps from 2 → 200). Auto-detection of dead stock, long-tail SKUs, and forecasted shortages.
- Cross-seller analytics (fastest selling categories, trending sizes, colour demand).
- Audit logs with timestamps, user IDs, IP tracking.
- System tools to enforce stock freezes on flagged SKUs.

---

## **PROBLEM 2: Customer Order Monitoring Dashboard**

### **For Fashion Orders with Multiple Variants & High Return Sensitivity Problem**

#### **Description**

Fashion orders require accurate tracking due to variant-level complexity (size, colour), packing stages, QA checks, and courier delays.

Build a comprehensive web dashboard that displays the entire lifecycle of an order—from placement to delivery—ensuring sellers and admins maintain full transparency and efficiency.

---

## **Seller Dashboard — Required Features**

- List of all incoming orders with SKU breakdowns.
- Ability to confirm/reject orders (with rejection reasons).
- Update for each stage: Accepted → Processing → QC Done → Packed → Shipped → Delivered.
- Automated packing slips & invoice generation.
- Ability to merge or split orders (important for multi-item fashion orders).
- Filters for date, SKU, size, fulfilment partner, delayed orders.
- Customer notes (measurements, fit preferences, delivery instructions).

## **Admin Dashboard — Required Features**

- Full visibility of all seller order flows.
  - SLA tracking: delays beyond 24/48/72 hours.
  - Alert system for high-risk orders (high value, COD, previous returns).
  - Ability to override order status if mis-updated.
  - Insights: order volume by category, seller performance, fulfilment bottlenecks.
  - Manual and automated QC checks for order activity.
- 

## **PROBLEM 3: Shipping Requests Reader & Label Generation Dashboard**

For Domestic/International Fashion Logistics Problem Description

Fashion orders involve complex delivery specs—international sizing compliance, fragile items, ironed garments, multi-piece sets. Shipping accuracy is critical.

Build a dashboard that reads customer shipping requests, validates addresses, and creates shipping labels.

---

## **Seller Dashboard — Required Features**

- View shipping requests auto-linked to order IDs.
- Address verification with error prompts (invalid pincode, incomplete street).
- Auto-generate PDF/PNG shipping labels with courier templates.
- Option to choose courier partner (FedEx, Delhivery, DHL).
- Pickup scheduling tool for courier pickups.
- Tracking ID integration & status panel.
- Ability to save commonly used shipping presets.

## **Admin Dashboard — Required Features**

- Monitor all shipping labels issued by sellers.
  - Address quality scoring system (low-confidence → flag).
  - Track courier performance through delivery metrics.
  - Ability to cancel or reassign courier partners for orders.
  - Export logs of all labels generated per seller, per day.
  - Dashboard heatmap of shipping destinations.
-

## **PROBLEM 4: Returns Management Dashboard**

For Fashion Returns with Defect Verification & Refund Flows Problem Description

Fashion has high return rates due to size/fit issues, colour mismatches, wrong items, and product defects.

Build a dashboard that helps sellers process returns smoothly while giving admins clear oversight and fraud control.

---

### **Seller Dashboard — Required Features**

- View all return requests with customer images/videos.
  - Full return status timeline (Requested → Review → Approved/Rejected → Item Received → QC Pass/Fail → Refunded/Replaced → Closed).
  - Approve/reject returns or request more evidence.
  - Mark item received and perform QC checks.
  - Issue refunds, replacements, store credits, or partial refunds.
  - Track causes of return (size, defect, colour mismatch, wrong item, etc.).
  - Access return analytics:
    - Return rate by SKU
    - Most common return reasons
    - High-return items flagged
    - Trend charts over time
- 

### **Admin Dashboard — Required Features**

- Oversight panel showing return statuses across all sellers.
  - Approve high-value or high-risk return cases.
  - Fraud detection (repeat abusers, mismatched evidence).
  - Enforce return policies and thresholds.
  - High-level analytics (category return rates, seller patterns).
  - Dispute resolution tools for escalated cases.
  -
-

## **PROBLEM 5: Product Catalog Dashboard**

For Adding, Managing & Maintaining Fashion Product Listings Problem Description

Fashion catalogs are media-heavy and variant-heavy. Managing products with size grids, colour swatches, style codes, material details, and lookbook images requires a robust interface. Build a product catalog dashboard where sellers manage their entire catalog and admins maintain platform-wide quality compliance.

---

### **Seller Dashboard — Required Features**

- Table/grid of all products with image thumbnails.
- Add new product with fields for:
  - Title, description
  - Pricing & discount
  - Colour, size, variant matrix
  - Fabric/composition details
  - Washcare details
  - Images & videos
- Bulk product upload.
- Archive, duplicate, or edit existing listings.
- Auto-image compression for faster upload.
- Category tagging (tops, dresses, menswear, footwear, accessories).

### **Admin Dashboard — Required Features**

- Approve/reject new listings or edits.
  - Auto-check for missing images, incorrect data, or banned items.
  - Compliance panel for material disclosure, category mapping, pricing rules.
  - Image quality inspection tools (blur, watermark detection).
  - Platform-wide product analytics (new launches, high performers).
  - Ability to edit descriptions to meet uniform standards.
-

## **FINANCIAL STABILITY & INVESTMENT EMPOWERMENT SUITE**

Hackathon Problem Statements — Web Dashboard Edition

Today's users face unprecedented financial challenges: inconsistent income, increasing digital spending, hidden debt traps, low long-term saving discipline, and difficulty starting investments.

Your task is to build five advanced web dashboards that help modern users build stability through automation, behaviour analysis, and personalised financial systems.

Each dashboard must include:

### **User Dashboard**

- Customisable settings
- Thresholds & rule builders
- Notifications & alerts
- Visual analytics
- Behaviour-based automations

### **Admin Dashboard**

- Oversight & monitoring
  - Risk detection
  - Compliance workflows
  - Insights & behavioural analytics
- 

## **1. MICRO-INVESTMENT & SPARE CHANGE ROUNDING DASHBOARD**

Helping today's users invest effortlessly through small daily actions

Expanded Problem Statement

Many users delay investing because they feel their income is too small or irregular. However, their everyday purchases create unused amounts that can be pooled and invested.

This dashboard automates micro-investing by rounding up daily transactions and investing the pooled amount into safe instruments—fully controlled by user-configurable settings.

---

## User Dashboard — Features & Customisation

### 1. Round-Up Settings

- Choose rounding level (₹5 / ₹10 / ₹20 / ₹50).
- Enable/disable daily round-ups.
- Set a minimum daily round-up amount.
- Set a maximum round-up limit to protect budgets.

### 2. Investment Bucket Controls

- Custom allocation across gold, ETFs, index funds, debt funds.
- Conditional rules:
  - “If today’s spare change > ₹25 → push entirely to gold.”
- Scheduled auto-rebalancing.

### 3. Threshold-Based Auto-Investment

- Set investment triggers (₹100 / ₹200 / ₹500 or custom).
- Choose weekly/monthly auto-invest cycles.

### 4. Notification Controls

- When spare change hits threshold.
- Low balance alerts.
- Failed deduction alerts.
- Streak and progress encouragements.

### 5. Visual Insights

- Growth timeline of spare change.
- Weekly and monthly summarised analytics.

---

## Admin Dashboard — Features

- Micro-investment flow monitoring.
  - Detection of suspicious or unusually high round-ups.
  - Popular bucket selections.
  - Round-up behaviour across segments.
  - Compliance logs for all micro-invest transactions.
- 
-

## **2. SIDE-HUSTLE INCOME TRACKER + SMART INVESTMENT PLANNER**

For users earning from multiple gigs, freelance payments, and side incomes

Expanded Problem Statement

Today's income patterns are fragmented: freelance projects, gig work, creator payouts, tips, commissions, and part-time earnings.

This dashboard must unify all income sources and automatically allocate a portion into investments based on user-defined rules and thresholds.

---

User Dashboard — Features & Customisation

### **1. Income Source Manager**

- Add multiple income types with custom labels.
- Set expected timelines for payments.
- Activate/deactivate sources.

### **2. Custom Rule Builder**

Rules examples:

- “Invest 10% of any payment above ₹1,500.”
- “Add fixed ₹200 to savings from every gig payment.”
- “Pause auto-invest if monthly income falls below ₹8,000.”

### **3. Threshold Controls**

- Monthly investment caps.
- Emergency pause triggers.
- Minimum income needed to trigger auto-invest.

### **4. Notification Settings**

- Payment received alerts.
- Rule-triggered investment alerts.
- Low-income month warnings.
- Stability reports.

### **5. Income Analytics**

- Income volatility meter.
  - Categorized income breakdown.
  - Investment consistency score.
-

## **Admin Dashboard — Features**

- Anonymised income trend analytics.
  - Alerts for prolonged low-income periods.
  - Rule usage analysis.
  - Detect repeated failed investments.
  - Compliance checks on large transactions.
- 
- 

## **3. DEBT HELPER DASHBOARD — BNPL, EMI & CREDIT OPTIMISER**

Helping today's users escape silent digital debt build-up

### Expanded Problem Statement

Today's users rely heavily on BNPL, micro-EMIs, credit lines, online shopping credit, and small-value loans.

Without visibility across all debt types, users often fall into high-interest traps.

This dashboard must consolidate all debts and provide customisable repayment strategies, early warnings, and repayment optimisation.

---

### User Dashboard — Features & Customisation

#### **1. Unified Debt Overview**

- Add or sync all debts.
- Custom tags per debt type.
- Adjustable due dates.

#### **2. “Danger Zone” Settings**

Users set their own:

- Overdue warning days
- Max EMI load
- Interest-rate alert threshold
- Credit card utilisation warning limit

### **3. Repayment Strategy Builder**

Users choose and customise:

- Snowball
- Avalanche
- Hybrid
- Custom weighted strategies

Rules like:

- “If salary crosses ₹25k → allocate extra ₹500 to highest interest debt.”

### **4. Notification Controls**

- Due date reminders
- Over-limit warnings
- Interest change alerts
- Missed payment alerts

### **5. Debt Health Score**

- Adjustable scoring logic
  - Improvement suggestions tailored to user patterns
- 

### **Admin Dashboard — Features**

- Risk-level heatmaps.
  - High-debt-load user detection.
  - BNPL dependency analytics.
  - Repayment success rates.
  - Regulatory compliance checks.
- 
-

## **4. GOAL-BASED INVESTING DASHBOARD — “MISSIONS” FOR TODAY’S USERS**

Turning savings into short-term, achievable, motivating challenges

Expanded Problem Statement

Today’s users prefer quick results and struggle to stay committed to long-term goals.

This dashboard must turn savings into missions with timelines, milestones, rewards, and motivational visuals, all fully customisable.

---

User Dashboard — Features & Customisation

### **1. Custom Goal Mission Builder**

- Build goals manually or pick templates.
- Set timeline (10–180 days).
- Choose saving frequency & amount.
- Add custom checkpoints.

### **2. Personalisation Settings**

- Rename goals.
- Select themes & styles.
- Upload vision/moodboard images.
- Set personal motivational messages.

### **3. Adaptive Suggestions**

- Timeline adjustment prompts.
- Recommendations based on progress.

### **4. Reward Engine**

- Custom badge triggers.
- Streak-based achievements.
- Optional shareable certificates.

### **5. Notification Settings**

- Daily nudges
- Streak updates
- Weekly progress reports
- Completion milestone alerts

---

Admin Dashboard — Features

- Popular goals & user patterns.
  - Drop-off analytics.
  - Mission success rates.
  - Ability to publish platform-wide challenges.
  - Suggest new templates based on performance.
- 
- 

## **5. REAL-TIME SPENDING BEHAVIOUR → INVESTMENT NUDGE ENGINE**

Converting impulse spending into meaningful investment actions

### **Expanded Problem Statement**

Today's users face overwhelming spending temptations: food delivery, impulsive shopping, subscriptions, micro-purchases.

This dashboard must detect spending patterns, identify unhealthy behaviour, and convert it into investment nudges powered by user-controlled rules and thresholds.

---

User Dashboard — Features & Customisation

### **1. Real-Time Spending Feed**

- Auto-categorise transactions.
- Custom category editing.
- Merge / rename categories.

### **2. Custom Spending Limits**

- Set daily, weekly, monthly budgets.
- Category-level spending caps.
- Customisable alert percentages (e.g., warn at 60%, 80%).
- Options for soft or hard limits.

### **3. Smart Nudge Controls**

Users set nudge style:

- Gentle
- Moderate
- Strong

Users create rules:

- “If food spend > ₹300/day → invest ₹50.”
- “Auto-invest 5% of all weekend spending.”
- “Convert unused subscription savings → investments.”

#### 4. Impulse Control Tools

- User-set cooldown periods.
- Spending lock windows (e.g., no shopping 10 PM–8 AM).
- Temporary category blocks.

#### 5. Spend-to-Invest Converter

- Set % of overspending to redirect as investment.
  - Weekly maximum cap.
  - Category-based enabling/disabling.
- 

#### Admin Dashboard — Features

- Spending trend analytics.
  - User behaviour clustering (food-heavy, impulsive buyers, etc.).
  - Nudge effectiveness tracking.
  - Pushable nudge templates.
  - Data privacy and usage compliance checks.
- 

### **Hackathon Problem Statement: Burnout & Focus Guardian**

Students, professionals, and founders frequently overload their schedules with long work hours, poor sleep, and meeting-heavy days—often realizing the cost only after burnout or missed goals. Existing tools track tasks and moods, but they fail to predict when performance collapse is about to happen or what to do about it.

Your challenge is to build Burnout & Focus Guardian, an AI-powered mobile application that predicts daily and weekly burnout or failure risk and generates clear, preventive actions. Using quick daily check-ins (sleep, work hours, mood, meetings, caffeine), the system should compute a risk score, identify key causes (e.g., sleep debt, overwork), and produce a one-tap action plan across Work, Health, and Focus.

The solution must go beyond tracking—acting as a risk radar + recovery planner. The MVP should be feasible within 24 hours and demonstrate how predictive insights can help users prevent burnout before it happens.

### **Hackathon Problem Statement: Finance Survival**

Most students and first-time earners struggle with real-world money management. Salaries disappear within days, emergency expenses destroy savings, and budgeting feels boring or overwhelming. Traditional finance apps focus on tracking numbers, not teaching decision-making under pressure—which is where most people fail.

Your challenge is to build Finance Survival, a simulation-based mobile game that teaches financial literacy through experience. Players receive a monthly salary, pay unavoidable expenses like rent and transport, make daily spending choices, and face random emergencies such as medical bills or repairs. Every decision impacts balance, health, and happiness.

The goal is simple: survive 30 days without going broke and end the month with savings. Poor planning leads to loss; smart budgeting leads to success and higher scores.

The solution must be engaging, logic-driven, and educational, demonstrating how game mechanics can teach budgeting, emergency planning, and needs-vs-wants thinking within a 24-hour hackathon build.

### **Hackathon Problem Statement: Smart Campus Issue Reporter**

College campuses face daily infrastructure issues such as power failures, water leakage, broken furniture, and network problems. Today, most of these issues are reported verbally or through informal channels like phone calls or messages. This leads to poor tracking, delayed resolution, repeated complaints, and a lack of accountability between students and campus administration.

Your challenge is to build a Smart Campus Issue Reporter App, a centralized digital platform that enables students to report campus issues easily and allows administrators to track, assign, and resolve them efficiently. The system should support issue categorization, location tagging, real-time status tracking, and basic analytics for administrators.

The goal is to improve transparency, reduce response time, and create a reliable history of campus maintenance issues. The solution must be simple, scalable, and feasible to build within a 24-hour hackathon, demonstrating how digital workflows can significantly improve campus operations and student experience.

### **Hackathon Problem Statement: Food Waste Reduction App**

Every day, large quantities of edible food are wasted by restaurants, hotels, hostels, and event organizers, while NGOs and needy communities struggle to access fresh meals. The absence of a real-time coordination system causes food to be discarded, leading to environmental damage, economic loss, and ethical concerns.

Your challenge is to build a Food Waste Reduction App that digitally connects food donors with NGOs and volunteers in real time. Donors should be able to post surplus food details such as type, quantity, freshness duration, and pickup location, while receivers can discover nearby available food, claim it quickly, and confirm collection.

The platform must ensure speed, transparency, and basic security while remaining simple enough to build within a 24-hour hackathon. The goal is to demonstrate how technology can reduce food waste, support vulnerable communities, and promote sustainable, community-driven food redistribution through an easy-to-use digital solution.

### **Hackathon Problem Statement: Smart Elder Care**

Millions of elderly people today live alone due to nuclear families and migration, making them vulnerable to missed medicines, unnoticed emergencies, and emotional isolation. Families often live far away and rely on frequent calls, yet critical moments can still go unnoticed. Existing health and senior-care apps are often complex, fragmented, or not designed for elderly usability.

Your challenge is to build Smart Elder Care, a simple, elder-friendly mobile application that acts as a daily digital caregiver. The app should combine medicine reminders, daily emotional check-ins, emergency SOS, and access to verified local caregivers into one unified platform. It must also provide a remote family dashboard for alerts, compliance tracking, and peace of mind.

The goal is to demonstrate how technology can proactively protect elderly users, reduce loneliness, and ensure faster emergency response. The solution should be scalable, intuitive, and feasible to prototype within a 24-hour hackathon while showcasing real social and commercial impact.

### **Hackathon Problem Statement: Verified Local Help-on-Demand**

Finding reliable local service workers such as electricians, plumbers, cleaners, or helpers remains risky and unstructured. Anyone can list themselves as a provider, leading to fake profiles, safety concerns, poor accountability, and inconsistent service quality. Households often rely on word-of-mouth or WhatsApp groups, while skilled workers struggle to gain steady, trusted employment.

Your challenge is to build Verified Local Help-on-Demand, a trust-first hyperlocal services platform that connects users with background-verified, rated, and insured local workers. The system should enable secure worker onboarding, hyperlocal discovery, simple job booking, and post-job ratings, while enforcing trust rules such as verification status, visibility limits, and complaint handling.

The goal is to demonstrate how digital trust layers—identity verification, reputation, and job protection—can make local hiring safer for households and more sustainable for workers. The solution should be practical, scalable, and achievable within a 24-hour hackathon MVP.

---

# 1. SOMEONE'S-CALLING DETECTOR LAMP

**A smart ambient lamp that helps users manage digital overwhelm and emotional boundaries.**

## Problem Statement

Today's users are overloaded with constant calls—work, family, toxic people, unknown numbers, spam, and even people they're avoiding. This leads to anxiety spikes, burnout, and poor emotional boundaries.

Build a smart desk or bedside lamp that glows differently depending on who is calling, helping users emotionally prepare, set boundaries, and reduce notification overload.

## What the IoT Device Should Do

- Bluetooth or WiFi connection to the phone.
- LED colour mapping based on contact category (user-customisable):
  - Red: People they're avoiding / toxic
  - Blue: Work
  - Pink: Crush or partner
  - Yellow: Close friends
  - Green: Family
- Pulse intensity corresponds to urgency (e.g., repeated calls).
- Mood-light mode when idle.

## User Dashboard Requirements

- Add, edit, create categories for contacts.
- Set colour & brightness for each category.
- Silence/do-not-disturb schedules per category.
- "Anxiety-free mode" → only certain categories glow.
- Call log visualised in categories with frequency charts.

## Admin Dashboard Requirements

- Monitor device performance across users (offline, error states).
- Track common category usage for product insights.
- Push firmware updates for LED behaviour and known bugs.
- Compliance view for privacy handling (no storing contact names).

---

Here is a cleaner, shorter, hackathon-ready version of the revised problem statement for Idea #2:

---

## 2. EMOTIONAL SUPPORT PLANT DEVICE

**A smart IoT add-on that gives your plant a personality and nudges users toward better self-care.**

### Problem Statement

Today's users often struggle with hydration, screen balance, and basic self-care routines. Plants help improve mood, but people forget to care for both themselves and their plants. Create a small IoT device that clips onto or sits inside any plant pot. It reacts to the user's lifestyle patterns through expressive lights and vibrations — making wellness fun, emotional, and interactive.

---

### Device Capabilities

- LEDs for mood expression (happy, calm, stressed, ignored, celebratory).
- Light & moisture sensing (optional soil sensor).
- Bluetooth/WiFi connection.
- Vibration cues for reminders (hydration, breaks, movement).

---

### User Dashboard Requirements

- Choose device personality (calm, chaotic, cute, sassy).
- Set rules: hydration reminders, screen-time alerts, step goals.
- View daily mood timeline of the device.
- Track hydration, wellness habits, and plant moisture/light levels.

---

### Admin Dashboard Requirements

- Device health monitoring (battery, connectivity).
  - Anonymous usage insights (popular rules, personalities).
  - Push firmware updates and new LED animations.
-

### 3. IoT PET OBJECT/ROCK WITH FEELINGS

**A humorous but psychologically meaningful device that rewards consistency.**

Problem Statement

Many users struggle with self-discipline, daily routines, and emotional consistency. Create a smart “pet object” that reacts emotionally to user engagement to build micro-habits—turning responsibility into fun and accountability into charm.

What the IoT Device Should Do

- LED eyes/face that animate emotions.
- Vibrations when it wants attention.
- Goes “sad mode” when ignored too long.
- “Happy mode” when user completes goals or interacts.
- Soft speaker for tiny beeps or giggles.

**User Dashboard Requirements**

- Customise object’s personality (chill, energetic, sarcastic).
- Choose daily engagements (2 taps, shake, talk-to-rock).
- Link habits (drink water, meditate, stretch) → object reacts.
- Streak tracker tied to object’s mood.
- Mood timeline (happy → neutral → dramatic → depressed → happy again).

**Admin Dashboard Requirements**

- Monitor object firmware/LED statuses.
  - Track which personalities are most used.
  - Analyse habit engagement trends (anonymous).
  - Push seasonal personality packs & animations.
- 

### 4. THE SNACK LOCK BOX

**A smart food-safe container that helps users balance cravings with wellbeing.**

Problem Statement

Users often struggle with binge-eating, late-night snacking, stress-eating, and inconsistent wellness habits.

Build a smart lock box for snacks that opens only under certain user-defined conditions—making healthy habits fun, automated, and behaviour-reinforcing.

What the IoT Device Should Do

- Secure locking mechanism with motorised latch.
- NFC/phone unlocking based on conditions.
- LED signals indicating “allowed”, “blocked”, “almost there”.
- Time-based or goal-based unlocking.

### **User Dashboard Requirements**

- Set unlock rules:
  - Hydration goals
  - Step count
  - Productivity minutes
  - Time of day
- “Cheat mode” with cooldown timer.
- Snack analytics (when, how often, emotional pattern).
- Emergency override with consequences (resets streak).

### **Admin Dashboard Requirements**

- Device reliability panel (motor health, battery).
  - Trend insights: most common unlock triggers.
  - Remote troubleshooting logs.
  - Behaviour usage data to improve rule templates.
- 

## **5. THE PROCRASTINATION SHOCK TILE**

### **A smart pressure-sensitive mat that keeps users at their workspace.**

#### Problem Statement

Users frequently procrastinate—walking away mid-task, pacing while stressed, getting distracted by random chores.

Create a pressure sensor floor tile/mat that detects when a user leaves their workspace and sends playful alerts or vibrations to nudge them back.

#### What the IoT Device Should Do

- Pressure sensor to detect presence/absence.
- Heatmap of movement patterns.
- Small vibration or audio cue when user steps off prematurely.
- “Focus mode” → gentle encouragement only.
- “Brutal honesty mode” → sassier audio or alerts.

### **User Dashboard Requirements**

- Set allowed break intervals.
- Define what counts as a legitimate break.
- Customise mat personality (calm, chaotic, sarcastic).
- View focus vs distraction timelines.
- Weekly focus score + movement visualisation.

## **Admin Dashboard Requirements**

- Device uptime & connectivity logs.
  - Sensor calibration tools.
  - Focus patterns across user cohorts.
  - Update packs for new sound/mode personalities.
- 

## **6. MARKETING CAMPAIGN RECOMMENDER**

Problem Statement:

Recommends optimal marketing campaigns tailored to customer segments based on historical performance and customer characteristics. Enables marketers to allocate budgets to high-ROI campaigns.

## **7. MARKETING CAMPAIGN CHANNEL RECOMMENDER**

Problem Statement

Identifies the most effective marketing channels (email, SMS, social media, etc.) for reaching specific customer segments. Maximizes engagement and conversion rates by matching audiences to preferred communication channels.

## **8. PROSPECT AND CUSTOMER SEGMENTATION**

Problem Statement:

Clusters customers and prospects into meaningful groups based on behavioral, demographic, and transactional attributes. Enables targeted strategies for different customer personas and lifecycle stages.

## **9. ESTIMATE CAMPAIGN EFFECTIVENESS**

Problem Statement:

Predicts the impact and ROI of marketing campaigns before or during execution. Measures key metrics like conversion rates, customer acquisition cost, and lifetime value attribution.

## **10. SMART PRICING ENGINE AND DISCOUNT RECOMMENDER**

Problem Statement:

Dynamically determines optimal pricing and discount strategies based on demand, inventory, competition, and customer elasticity. Maximizes revenue and profit margins while maintaining competitiveness.

## **11. FORECAST SALES**

Problem Statement:

Predicts future sales volumes across products, regions, or time periods using historical data and trend analysis. Enables inventory planning, resource allocation, and revenue forecasting.

## **12. DETERMINE THE BEST PRODUCT TO BE SOLD TO A PROSPECT**

Problem Statement:

Recommends the most suitable products for individual prospects based on their profile, purchase history, and needs. Increases cross-sell success rates and customer satisfaction.

## **13. DETERMINE THE BEST PROSPECT TO SELL TO**

Problem Statement:

Identifies and ranks prospects with the highest propensity to purchase or convert. Optimizes sales team effort allocation and improves closing rates.

## **14. CROSS SELLING AND UPSELLING RECOMMENDER**

Problem Statement:

Suggests complementary products and premium upgrades to existing customers based on purchase patterns and preferences. Increases average order value and customer lifetime value.

## **15. REAL-TIME PAYMENT FRAUD DETECTION**

Problem Statement:

Detects fraudulent transactions in real-time using anomaly detection and pattern recognition on payment attributes. Minimizes financial losses and protects customer trust.

## **16. ESTIMATE ORDER VOLUME**

Problem Statement:

Forecasts expected order quantities across products and time periods using demand modeling and trend analysis. Optimizes inventory levels and production planning.

## **17. EMPLOYEE ATTRITION PREDICTION**

Problem Statement:

Predicts which employees are at high risk of leaving based on engagement, compensation, and tenure data. Enables targeted retention programs and succession planning.

## **18. PREDICT FULFILMENT DELAYS**

Problem Statement:

Identifies orders at risk of delayed fulfillment based on operational and supply chain factors. Enables proactive customer communication and resource allocation adjustments.

## **19. ANTI-MONEY LAUNDERING TRANSACTION TYPOLOGY DISCOVERY**

Problem Statement:

Identifies suspicious transaction patterns and typologies indicative of money laundering activities. Enables compliance teams to flag high-risk transactions for investigation.

## **20. CLAIMS FRAUD AND TRIAGE AUTOMATION**

Problem Statement:

Detects fraudulent insurance claims and automatically prioritizes them for investigation or approval. Reduces claim processing time and fraud losses.