Printos: An Integrated Web-Based Solution for Print-On-Demand Apparel Businesses

Vedant Deepak Pawar[1], Dhawal Subash Pawar[2], Sahil Narendra Dandgawhal[3], Kumudini Wagh[4], Prof. D. M. Deshpande[5]

[1,2,3,4]*Department of Computer Engineering, Gokhale Education Society's R. H. Sapat College of Engineering, Nashik*

[5]*Guide, Department of Computer Engineering, Gokhale Education Society's R. H. Sapat College of Engineering, Nashik*

***Abstract*—Small-scale Print-on-Demand (POD) businesses face significant operational challenges due to fragmented workflows, manual inventory management, and inaccurate costing. PrintOS is a comprehensive web-based Single Page Application (SPA) that integrates product customization with real-time dynamic cost calculation, intelligent inventory management, and an automated print layout generator utilizing two-dimensional bin packing algorithms. Built entirely on the client-side using HTML5, CSS, and JavaScript, PrintOS requires zero backend infrastructure while providing powerful business intelligence. A rule-based AI advisor provides intelligent restock recommendations preventing both stockouts and overstock situations. The system leverages foundational principles from Lean Manufacturing, Activity Based Costing, and computational geometry to deliver significant improvements in operational efficiency, material waste reduction, and profit margin accuracy. This paper presents the architecture, design methodologies, implementation details, and comprehensive evaluation of PrintOS, demonstrating its effectiveness as a tool for empowering small entrepreneurs in the custom apparel industry.**

***Index Terms*—Print-on-Demand, Web Application, Business Management, Inventory Optimization, Two-Dimensional Bin Packing, Dynamic Costing**

I. INTRODUCTION

A. Motivation

The custom apparel industry in India has experienced explosive growth over the past five years, particularly in urban centers like Nashik. The rise of e-commerce, social media marketing, and event-based merchandise has created a burgeoning market for Print-on-Demand (POD) businesses. However, this growth is constrained by a critical operational bottleneck: the absence of affordable, integrated software solutions designed specifically for small-scale entrepreneurs. Current workflows are highly fragmented. Entrepreneurs typically utilize multiple disconnected tools:

- Adobe Photoshop or similar for design
- Microsoft Excel for inventory tracking
- Manual processes for production layout
- Paper-based or scattered digital records for financial tracking

This fragmented approach introduces multiple failure points:

1) Redundant Data Entry: Information is entered into multiple systems, creating inconsistencies.
2) Material Waste: Manual arrangement of logos onto print sheets results in significant waste of expensive transfer paper and ink.
3) Inaccurate Costing: Pricing decisions are often based on intuition rather than data, leading to inconsistent profit margins.
4) Poor Inventory Management: Spreadsheet-based systems are error-prone and lack real-time updates, resulting in stockouts or overstock situations.

PrintOS directly addresses these challenges by providing an integrated platform that consolidates design, costing, inventory, and production into a single, intuitive system.

B. Problem Definition

Small-scale Print-on-Demand businesses lack an integrated, affordable software solution that consolidates their core operational workflows. The primary problems are:

1) Workflow Fragmentation: Disconnected applications lead to redundant data entry, high error rates, and operational inefficiency.
2) Production Inefficiency: Manual logo arrangement on large format sheets (A3: 11.7” × 16.5”) is time-consuming and results in significant material waste.
3) Costing Inaccuracy: Without activity-based costing methodologies, entrepreneurs cannot accurately correlate logo size to material costs, resulting in suboptimal pricing.
4) Inventory Mismanagement: Spreadsheet-based tracking lacks real-time updates and predictive capabilities, leading to both stockouts and overstock situations.
5) Poor Business Visibility: Entrepreneurs lack real-time visibility into financial metrics (revenue, cost, profit) necessary for strategic decision-making.

II. LITERATURE SURVEY

A. Existing Print-on-Demand Software Solutions

The market for print-on-demand software is dominated by two categories of solutions:

E-Commerce Plugins: Solutions like Printful and Printify serve as integration layers between e-commerce platforms (Shopify, WooCommerce) and production facilities. These solutions excel at order fulfilment and logistics but are fundamentally designed for businesses that outsource manufacturing. They charge per-order fees and provide minimal control over the production process, making them unsuitable for entrepreneurs managing their own printing operations.

Standalone Design Tools: Tools like Adobe Illustrator and Placeit are powerful design applications but lack business intelligence, inventory management, and production optimization features. They excel at creative output but provide no support for the operational side of the business.

Conclusion: No integrated, affordable solution exists for entrepreneurs who handle their own manufacturing and inventory management.

B. Two-Dimensional Bin Packing

The problem of efficiently arranging logos onto a fixed canvas (the A3 sheet) is a classical computer science problem:

The Two-Dimensional Bin Packing Problem (2D-BP).

Problem Formulation: Given a rectangular bin of dimensions $W \times H$ (in our case, an A3 sheet: 3508 × 4961 pixels at 300 DPI) and a set of rectangular items (logos) with dimensions (w_i, h_i) and quantities q_i, the objective is to arrange all items into the minimum number of bins.

Problem Complexity: The optimal solution to 2D-BP is NP-Hard. Exhaustive search of all possible arrangements is computationally infeasible for realistic problem instances. Therefore, heuristic approaches are required.

Our Implementation - Greedy Algorithm: PrintOS implements a simplified greedy algorithm (First Fit Decreasing Height - FFDH heuristic):

1) Sort items by descending height
2) For each item, place it in the first available position that fits
3) If no position available in current level, create new level 4) Continue until all items are placed

This algorithm runs in $O(n\log n)$ time and provides solutions that are typically within 11-15% of optimal. For the domain of print production, this” good enough” solution provides substantial improvement over manual layout while maintaining computational efficiency.

C. Lean Manufacturing and Just-In-Time Inventory

The concept of” lean” inventory—maintaining minimum stock levels while avoiding stockouts—originates from the Toyota Production System. Key principles include:

- Eliminate Waste: Overstock ties up capital and increases risk of product obsolescence.
- Just-in-Time (JIT): Inventory should be available exactly when needed, neither before nor after.
- Continuous Improvement: Regular monitoring and adjustment of inventory levels based on actual demand.

PrintOS applies these principles through an AI Restock Advisor that:

1) Monitors inventory levels in real-time
2) Tracks sales velocity for each product-size combination
3) Generates alerts when inventory falls below a configurable threshold

4) Provides data-driven recommendations for restock quantities

D. Activity-Based Costing (ABC)

Traditional costing models assign overhead uniformly across all products. Activity-Based Costing, introduced by Kaplan and Cooper, assigns costs based on the" activities" that drive consumption of resources.

In the context of print production:

- Primary Cost Driver: The size of the logo (measured in square pixels or physical dimensions).
- Resource Consumption: Larger logos consume more ink and transfer paper.
- Accurate Pricing: By tying cost to logo size, PrintOS ensures that each item is priced to deliver the target profit margin.

PrintOS Cost Model:

$$PrintCost = k \times LogoArea \times MaterialCost$$

$$SalePrice = \frac{PrintCost}{1 - TargetMargin}$$

III. SYSTEM ARCHITECTURE

A. High-Level Architecture

PrintOS is architected as a Client-Side Single Page Application (SPA). This design choice offers several advantages:

- Zero Infrastructure Costs: No backend server or database is required.
- Rapid Development: All business logic can be implemented in JavaScript.
- Data Privacy: All user data remains on the user's device.
- Offline Capability: The application functions without internet connectivity.
- Immediate Deployment: No deployment pipeline or DevOps complexity.

B. Presentation Layer

Built using HTML5 and CSS (with Tailwind CSS for styling), the presentation layer provides:

- Product customizer interface with drag-and-drop logo placement
- Real-time canvas preview showing products with logos
- Financial dashboard displaying key metrics
- Inventory management controls
- Print file generation interface
- Responsive design for multiple screen sizes

C. Business Logic Layer

Implemented in vanilla JavaScript, this layer handles:

- User interactions and event handling
- Canvas rendering and logo manipulation
- Dynamic cost calculations based on logo size
- Inventory state management and validation
- Two-dimensional bin packing algorithm
- Financial metrics computation (revenue, cost, profit, margins)
- Restock recommendation generation

D. Data Layer

All application state is stored in JavaScript objects within the browser's memory:

- Inventory data: Stock levels and sold quantities for each product-size combination
- Financial data: Revenue, costs, profit, margins
- Product configurations: Base prices, sizes, materials
- Sales transaction history: Complete record of all transactions with timestamps

E. System Architecture Overview

The three layers work together as follows:

1) User interacts with Presentation Layer (HTML/CSS)
2) Interactions trigger events in Business Logic Layer (JavaScript)
3) Business Logic performs calculations and updates Data Layer
4) Changes in Data Layer trigger re-rendering of Presentation Layer
5) Real-time updates provide immediate feedback to user

IV. SYSTEM DESIGN

A. Functional Architecture

PrintOS consists of five core functional modules:

1) Module 1: Product Customizer: Features:

- Logo Upload: Users upload logo images (PNG, JPG, GIF)

- Product Selection: Choose between t-shirt, hoodie, or other apparel
- Size Selection: Select preview size (L, XL, XXL)
- Logo Manipulation: Adjust logo position (X, Y) and size using interactive sliders
- Canvas Rendering: Real-time preview on HTML5 Canvas showing how logo appears on product

User can see changes in real-time as they adjust size and position.

2) Module 2: Dynamic Cost Calculator: Features:
- Logo Size Analysis: Calculate logo area in pixels
- Material Cost Mapping: Map logo area to ink and paper consumption
- Cost Calculation: Compute print cost based on activity-based costing principles
- Sale Price Calculation: Apply target profit margin to determine selling price

Cost Calculation Formula:

$$SalePrice = \frac{PrintCost}{1 - TargetMargin}$$

Where *PrintCost* is calculated from logo size and material costs, and *Target Margin* is the desired profit margin (typically 0.40 for 40%).

3) Module 3: Inventory Management: Features:
- Stock Tracking: Monitor inventory levels for each product-size combination
- Sales Recording: Decrement inventory upon sale
- Manual Updates: Adjust inventory levels when new stock arrives
- Validation: Prevent overselling by checking stock availability before recording sale

4) Module 4: Financial Dashboard: Features:
- Revenue Tracking: Sum of all sales prices
- Cost Tracking: Sum of all print costs
- Profit Calculation: Revenue minus Cost
- Profitability Metrics: Profit margin, profit per unit, ROI

Displays real-time metrics that update with each sale.

5) Module 5: Print Layout Generator: Features:
- Batch Optimization: Arrange all logos from an order onto A3 sheets
- 2D Bin Packing: Utilize the greedy algorithm to maximize material utilization
- Canvas Rendering: Generate high-resolution PNG files (300 DPI)
- File Export: Provide download-ready files for production

Uses the 2D bin packing algorithm to arrange multiple logos efficiently on A3 sheets.

V. METHODOLOGY

A. Development Approach

PrintOS was developed using an iterative, agile methodology with emphasis on user feedback and incremental feature delivery.

1) Phase 1: Requirements Analysis:
- User interviews with small business owners
- Identification of core pain points
- Definition of functional and non-functional requirements
- System architecture planning

2) Phase 2: Foundation Implementation:
- HTML structure and CSS styling
- Basic UI shell with navigation
- State management framework

3) Phase 3: Core Module Development:
- Product customizer with canvas rendering
- Logo upload and manipulation
- Dynamic cost calculator • Inventory management system

4) Phase 4: Advanced Features:
- Two-dimensional bin packing algorithm
- AI restock advisor
- Financial dashboard
- Print file generation

5) Phase 5: Testing and Optimization:
- Unit testing of business logic
- User acceptance testing with real business owners
- Performance optimization
- Bug fixes and refinement

B. Algorithm: 2D Bin Packing

Algorithm:
1. Sort items by descending height
2. Create first bin
3. For each item:

- Check if it fits in current level of current bin
- If yes, place it in first available position
- If no, create new level in bin or move to next bin

4. Return array of packed bins

- Time Complexity: $O(n \log n)$
- Space Complexity: $O(n)$

C. Algorithm: AI Restock Advisor

Algorithm:

1. Calculate average daily sales (7-day moving average) for each product
2. Calculate days of supply: $\text{Current Stock} \div \text{Average Daily Sales}$
3. If days of supply <3: Mark as CRITICAL (HIGH urgency)
4. If days of supply <7: Mark as WARNING (MEDIUM urgency)
5. Generate recommendations for reordering

This simple rule-based approach effectively prevents both stockouts and overstock.

VI. IMPLEMENTATION

A. Technology Stack

Presentation Layer:

- HTML5: Semantic markup for structure
- CSS3 + Tailwind CSS: Responsive styling
- JavaScript (ES6+): User interaction handling

Canvas Layer:

- HTML5 Canvas API: Product preview rendering
- File API: Logo file handling

State Management:

- Vanilla JavaScript objects: State storage

B. Core Implementation Details

```
class ProductCustomizer {
constructor(canvasId, mockupImages) {
this.canvas = document.getElementById(canvasId);
this.ctx = this.canvas.getContext('2d');
this.mockupImages = mockupImages;
this.logoConfig = { x: 0, y: 0, size: 100, scale: 1 };
  }

renderPreview(product, size) {
const mockup = this.mockupImages[product][size];
this.ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);
this.ctx.drawImage(mockup, 0, 0);

if (this.logoImage) {
const logoDim = this.calculateLogoDimensions();
this.ctx.drawImage(this.logoImage,
this.logoConfig.x, this.logoConfig.y, logoDim.width, logoDim.height);
  }
   }

calculateLogoDimensions() {
const baseSize = this.logoConfig.size;
const scale = this.logoConfig.scale;
const aspectRatio = this.logoImage.width / this.logoImage.height;
return { width: baseSize * scale, height: (baseSize * scale) / aspectRatio };
  }
}


class CostCalculator {
constructor(baseRate = 0.05, targetMargin = 0.40) {
this.baseRate = baseRate;
this.targetMargin = targetMargin;
 }

calculatePrintCost(logoDimensions, dpi = 300) {
const logoPxArea = logoDimensions.width * logoDimensions.height;
const logoInArea = logoPxArea / (dpi * dpi);
const materialCost = logoInArea * this.baseRate;
const fixedOverhead = 0.50;
return materialCost + fixedOverhead;
}

calculateSalePrice(printCost) {
return printCost / (1 - this.targetMargin);
  }
}

class InventoryManager {
constructor() {
this.inventory = {
'tshirt': { 'L': 50, 'XL': 40, 'XXL': 30 },
'hoodie': { 'L': 25, 'XL': 20, 'XXL': 15 }
      };
```

```
this.salesHistory = [];
    }

recordSale(product, size, quantity, cost, price) {
if (this.inventory[product][size] < quantity) {
throw new Error('Insufficient stock');
    }
this.inventory[product][size] -= quantity;
const transaction = {
product,
size,
quantity,
cost,
price,
timestamp: new Date(),
profit: (price - cost) * quantity
  };
this.salesHistory.push(transaction);
return transaction;
  }

updateStock(product, size, quantity) {
this.inventory[product][size] = Math.max(0,
this.inventory[product][size] + quantity);
  }
}
```

VII. RESULTS AND EVALUATION

A. Performance Metrics

1) Canvas Rendering Performance:

- Logo upload and decode: 45 ± 8 ms
- Canvas redraws (single logo): 2 ± 0.5 ms
- Canvas redraws (10 logos): 8 ± 1 ms
- Logo resize/reposition: < 1 ms

Result: Logo manipulation is responsive with latency < 100ms, meeting real-time interaction requirements.

Order Size	Sheets Used	Utilization	Time (ms)
10 items	1	87%	12
25 items	2	85%	18
50 items	3	83%	31
100 items	5	82%	54

2) 2D Bin Packing Efficiency:

Result: The greedy algorithm achieves 82-87% material utilization, representing approximately 25-30% waste reduction compared to manual layout.

Logo Size	Measured Cost	Calculated Cost	Error
Small (500px^2)	\$0.82	\$0.79	-3.7%
Medium (1000px^2)	\$1.35	\$1.33	-1.5%
Large (2000px^2)	\$2.18	\$2.21	+1.4%

3) Cost Calculation Accuracy:

Result: Calculated costs are within ±4% of measured values, sufficient for pricing decisions.

B. Business Impact Assessment

Conducted a 4-week pilot with 5 small business owners: *1) Efficiency Gains:*

- Average order processing time reduced by 65% (from 45 minutes to 16 minutes)
- Manual file creation time eliminated entirely
- Number of data entry errors reduced from 8-10 per week to 0-1 per week 2) Cost Savings:
- Material waste reduced by 28% on average
- Transfer paper cost savings: \$120-\$180 per month per business
- Labor time savings equivalent to 3-5 hours per week 3) Revenue Improvements:
- Pricing accuracy improved, increasing profit margins by 8-12%
- Reduced stockouts improved sales by 3-5%
- Better inventory visibility enabled more informed product mix decisions

C. User Satisfaction

Post-pilot survey results (n = 5, Likert scale 1-5):

- Ease of Use: 4.6
- Workflow Improvement: 4.8
- Feature Completeness: 4.4
- Value for Cost: 5.0
- Overall Satisfaction: 4.7

Qualitative Feedback:” PrintOS has transformed how I manage my business. I used to spend 4-5 hours per week on inventory and production planning. Now it’s automated and accurate. My profit margins have improved noticeably because I’m pricing correctly.” — Business Owner, Pilot Study

VIII. DISCUSSION

A. Key Contributions

1) Integrated Platform: First integrated solution specifically designed for small POD businesses managing their own manufacturing.
2) Activity-Based Costing in Web Apps: Practical implementation of ABC principles for real-time pricing in a web application context.
3) Computational Efficiency: Demonstrates that nontrivial algorithms (2D bin packing) can be effectively implemented in client-side JavaScript while maintaining excellent performance.
4) Zero-Infrastructure Model: Shows that complete business management applications can be delivered as pure client-side SPAs, enabling rapid deployment and zero IT overhead.
5) Data-Driven Inventory Management: Implements lean manufacturing principles in an accessible, user-friendly system suitable for non-technical entrepreneurs.

B. Limitations

1) Data Persistence: All data is lost when the browser is closed or refreshed. A backend database would enable persistent storage and multi-device access.
2) Single User: The system does not support multiple user accounts or cloud-based saving of designs.
3) No Real-Time Collaboration: Multiple team members cannot work simultaneously on the same order.
4) Limited Reporting: The dashboard provides basic metrics but lacks advanced analytics like sales forecasting or trend analysis.
5) Manual Integration: Orders must be manually entered; no integration with e-commerce platforms (Shopify, WooCommerce).
6) Static 2D Packing: The bin packing algorithm does not consider product rotation, which could improve utilization in some cases.

IX. FUTURE WORK

A. Short-Term Enhancements

1) Persistent Storage:

Implement browser LocalStorage API to persist session data across browser sessions, providing a basic form of data durability without requiring backend infrastructure.

2) Advanced Packing Algorithm: Enhance the 2D bin packing algorithm to:

- Support item rotation (for non-square logos)
- Implement guillotine cuts for improved space utilization
- Use bottom-left decreasing height heuristic 3) Enhanced Dashboard:
- Interactive charts using Chart.js or similar
- Sales trend analysis
- Inventory velocity charts
- Profitability by product analysis

B. Medium-Term Roadmap

1) Cloud Backend: Migrate to a full-stack architecture (MERN or Firebase) to support:

- Persistent data storage
- Multi-user accounts with access control
- Cloud-based design library
- Cross-device synchronization 2) E-commerce Integration:
- Shopify API integration for automatic order import
- WooCommerce plugin development • Real-time order status updates to e-commerce platform 3) AI/ML Enhancements:
- Machine learning-based demand forecasting
- Intelligent pricing optimization
- Anomaly detection for inventory discrepancies
- Recommendation engine for product mix optimization

C. Long-Term Vision

1) Manufacturing Partner Integration: APIs enabling direct integration with print production partners for automated order fulfillment.
2) Supply Chain Management: Extend system to manage supplier relationships, procurement, and payment workflows.
3) Marketplace Ecosystem: Create a platform connecting multiple small businesses with shared manufacturing resources.
4) Mobile Application: Native iOS and Android apps for on-the-go business management.
5) AI Assistant: Conversational AI interface for natural language business queries and recommendations.

X. CONCLUSION

PrintOS successfully demonstrates how modern web technologies can address real business problems for small entrepreneurs in the custom apparel industry. By integrating product customization, dynamic cost calculation, intelligent inventory management, and automated production optimization into a single, accessible platform, PrintOS provides immediate, measurable business value. The system's client-side architecture eliminates deployment complexity and infrastructure costs, making it economically accessible to small business owners.

The implementation of proven business principles—Activity-Based Costing, Lean Manufacturing, and computational geometry—in an intuitive interface shows that sophisticated business logic can be effectively delivered to non-technical users. With pilot-tested improvements in operational efficiency (65% faster order processing), material waste reduction (28%), and profitability improvements (812% margin increase), PrintOS has demonstrated compelling business value. User satisfaction scores of 4.6+ across all dimensions indicate strong product-market fit.

PrintOS represents a significant step forward in democratizing business management technology for India's growing entrepreneurial ecosystem. By removing operational complexity as a barrier to entry, PrintOS enables entrepreneurs to focus on what they do best: creating quality products and building their brands. The modular architecture and clear separation of concerns provide a solid foundation for future enhancement. The roadmap toward cloud-based persistence, ecommerce integration, and AI-powered decision support will further extend the platform's capabilities and market reach.

REFERENCES

[1] Lodi, S. Martello, and M. Monaci, "A new heuristic for two-dimensional bin packing," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 219–235, 2002.

[2] J. K. Liker, *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. Boca Raton, FL, USA: Productivity Press, 2004.

[3] R. S. Kaplan and R. Cooper, *Cost and Effect: Using Integrated Cost Systems to Drive Profitability and Performance*. Boston, MA, USA: Harvard Business Press, 1998.

[4] V. A. Mabert, A. Soni, and M. A. Venkataramanan, "Enterprise resource planning: Common myths versus evolving reality," *Business Horizons*, vol. 44, no. 1, pp. 69–76, 2001.

[5] Mozilla Developer Network, "Canvas API," 2024.

[6] Tailwind CSS, "Official Documentation," 2024.

[7] ECMA International, *ECMAScript 2015 Language Specification*, ECMA-262, 6th ed., 2015.