

PrintSmart - College Print Order & Pickup System
A Project Report
Submitted
In Partial Fulfillment of the Requirements
For the Degree of

Bachelor of Technology (B.Tech)
in
Computer Science Engineering(Data Science)
by

Prince Singh
2301921540131

Mrityunjay Kumar Thakur
2301921540110

Rammohan Chauhan
2301921540146

Pratik Raj
2301921540129

Under the Supervision of
Dr. Rituparna Sarma
Assistant Professor



G.L BAJAJ INSTITUTE OF TECHNOLOGY AND MANAGEMENT
GREATER NOIDA



DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY,
UTTAR PRADESH, LUCKNOW
DECEMBER 2025

DECLARATION

We hereby declare that the project work presented in this report entitled “PrintSmart - College Print Order & Pickup System”, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science Engineering (Data Science), submitted to Dr. A. P. J. Abdul Kalam Technical University, Uttar Pradesh , Lucknow is based on our own work carried out at the Department of Computer Science & Engineering(Data Science), G.L. Bajaj Institute of Technology and Management, Greater Noida. The work contained in the report is true and original to the best of our knowledge, and the project work reported in this report has not been submitted by us for the award of any other degree or diploma.

Signature:

Name : PRINCE SINGH

Roll No: 2301921540131

Signature:

Name : MRITYUNJAY KUMAR THAKUR

Roll No: 2301921540110

Signature:

Name : RAMMOHAN CHAUHAN

Roll No: 2301921540146

Signature:

Name : PRATIK RAJ

Roll No: 2301921540129

Date: 06/12/2025

Place: Greater Noida

CERTIFICATE

This is to certify that the Project report entitled “PrintSmart - College Print Order & Pickup System” done by Prince Singh (2301921540131), Mrityunjay Kumar Thakur (2301921540110), Rammohan Chauhan (2301921540146) and Pratik Raj (2301921540129) is an original work carried out by them in Department of Computer Science & Engineering(Data Science), G.L. Bajaj Institute of Technology and Management, Greater Noida, under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma, to the best of my knowledge and belief.

Date: 06/12/2025

Dr.Rituparna Sarma

Signature of the Supervisor

Dr. Harikesh Singh

Head of the Department

ACKNOWLEDGEMENT

The merciful guidance bestowed to us by the Almighty helped us complete this project successfully. We humbly pray with sincere hearts for His guidance to continue forever.

We thank our project guide, Dr. Rituparna Sarma, for her guidance and support during this project. Her versatile knowledge has helped us during critical times during the span of this project.

We pay special thanks to our Head of the Department Dr. Harikesh Singh who has been always present as a support and help us in all possible ways during this project.

We also take this opportunity to express our gratitude to all those who have been directly and indirectly involved with us during the completion of the project.

We would like to thank our friends who have always encouraged us during this project.

Last but not least, I would like to thank all the faculty members of the Computer Science and Engineering(Data Science) department who provided valuable suggestions during the project.

ABSTRACT

Printing services remain an essential component of academic institutions; however, the conventional workflow used in most college campuses is predominantly manual, time-consuming, and prone to operational inefficiencies. Students often depend on physical queues, unstructured communication channels, and unreliable file-sharing methods, resulting in delays and frequent mismanagement of their time. This project presents *PrintSmart*, a cloud-based, serverless printing management system designed to streamline and automate the end-to-end print ordering process. The system leverages Firebase Authentication for secure role-based access control, Firebase Realtime Database for instantaneous synchronization between users, and Cloudinary for optimized document and payment proof storage. Students can upload files, configure print settings, make payments through UPI intent, and track order status in real time, while shop owners obtain a structured dashboard for efficient order handling using a first-come-first-served model. The proposed solution significantly reduces queue congestion, minimizes communication errors, enhances workflow transparency, and improves overall service efficiency in academic environments. The outcomes demonstrate that a lightweight, scalable, and serverless architecture can modernize traditional campus printing operations and serve as a foundation for future extensions, including automated UPI verification, OCR-based analytics, and multi-shop integration.

TABLE OF CONTENT

Declaration	(2)
Certificate	(3)
Acknowledgement	(4)
Abstract	(5)
Table of content	(6)
List of figures	(7)
List of tables	(8)
List of abbreviations	(9)
Chapter 1. Introduction	(10)
1.1 Preliminaries	
1.2 Motivation	
1.3 Project Overview/Project Specification	
1.4 Aim and objectives	
Chapter 2. Literature Survey	(14)
2.1 Introduction	
2.2 Existing System	
2.3 Benefits of the project	
Chapter 3. Proposed Methodology	(19)
3.1 Problem Formulation	
3.2 System Analysis & Design	
3.3 Proposed Work	
Chapter 4. Implementation	(28)
4.1 Introduction	
4.2 Implementation Strategy (Flowchart,Algorithm etc.)	
4.3 Tools/Hardware/Software requirements	
4.4 Expected Outcome	
Chapter 5. Result & Discussion	(33)
5.1 System Results	
5.2 Discussion of findings	
5.3 Performance Evaluation	
Chapter 6. Conclusion and future scope	(39)
6.1 Conclusion	
6.2 Future Scope	
6.3 Summary	
References : Appendix I: Plagiarism Report of Project Report (<=15%)	

LIST OF FIGURES

Figure No.	Description	Page No.
Figure 1.1	High Level Architecture Diagram	12
Figure 3.1	System Architecture of PrintSmart	21
Figure 3.2	Firebase Realtime Database Structure	22
Figure 3.3	UI Flow Diagram	23
Figure 5.1	Student Dashboard	33
Figure 5.2	Shop Owner's Dashboard	35

LIST OF TABLES

Table No.	Description	Page No.
Table 4.1	Software Requirements	30
Table 4.2	Hardware Requirements	30
Table 5.1	Performance Metrics	37

LIST OF ABBREVIATIONS

Abbreviation	Full Form	Page No.
UPI	Unified Payment Interface	11
UI	User Interface	20
PDF	Portable Document Format	21
FCFS	First-Come First-Served	30
API	Application Programming Interface	38
OCR	Optical Character Recognition	40
URL	Uniform Resource Locator	22
HTML	Hyper Text Markup Language	30
CSS	Cascading Style Sheets	30
JSON	Javascript Object Notation	12
PSP	Payment Service Provider	16

Chapter 1

INTRODUCTION

1.1 Preliminaries

Printing facilities constitute an essential support service within higher education institutions, as students routinely require printed copies of assignments, laboratory records, reports, project documentation, diagrams, and various academic materials. Traditionally, this process has relied heavily on manual interactions: students physically visit print shops, wait in queues, share documents using email or messaging platforms, verbally communicate printing requirements, make payments, and then wait for their turn to receive the printed copies. Although this practice has been entrenched in campus environments for years, it increasingly reveals its limitations.

With rising student populations and increasing academic workloads, the shortcomings of this manual workflow have become more apparent. During peak submission periods, print shops often face overcrowding, leading to extended waiting times, misplaced files, and frequent miscommunication. The absence of a centralized digital mechanism exacerbates inefficiencies for both students and shop operators.

Advances in cloud computing, real-time databases, web-based automation, and digital payment systems present an opportunity to transform this outdated model. PrintSmart was conceived to meet this need by offering a unified digital platform through which students can upload documents, configure printing preferences, make payments, and track order progress in real time, while enabling shop owners to manage requests through an organized, automated dashboard.

1.2 Motivation

The inception of PrintSmart stems from practical observations of the challenges students face during peak printing hours at academic institutions. Long queues consume valuable time, while reliance on platforms such as WhatsApp or email for file sharing often leads to lost documents, misinterpreted instructions, and overall inefficiency. Shop owners also experience operational difficulties due to the absence of structured workflows; miscommunication, skipped orders, and unintentional reordering are common.

Additionally, higher education institutions are increasingly adopting digital transformation initiatives. A centralized, automated print-ordering system aligns with this broader push toward modernization. Such a solution can streamline file submission, eliminate unnecessary waiting, and improve transparency for students.

From the shop owner's perspective, a structured dashboard that organizes tasks, displays file links, and provides clear print instructions significantly reduces operational errors. PrintSmart addresses these issues while establishing a foundation for future integrations such as automated UPI validation, OCR-enabled analytics, and multi-shop support. The aim is not merely to create another web tool, but to introduce a scalable system capable of real-world deployment across diverse academic environments.

1.3 Project Overview / Project Specifications

PrintSmart is a cloud-driven print management solution designed to digitally connect students and print shop owners. It automates the entire lifecycle of a print request—from document upload and print configuration to payment verification and order pickup.

Key System Features

1. Student Portal

- Upload documents via Cloudinary
- Set printing preferences
- Automatically calculate printing costs
- Initiate UPI-based payments
- Upload payment confirmation screenshots
- Track printing status in real time

2. Shop Owner Portal

- Receive and view incoming orders instantly
- Access document links and payment proofs
- Update the status of print orders
- Follow FCFS (First-Come-First-Served) processing
- Manage workload through a structured dashboard

3. Cloud Services Utilized

- **Firebase Authentication:** Secured login and role management
- **Firebase Realtime Database:** Live synchronization of print orders
- **Cloudinary:** Storage of uploaded documents and payment screenshots

4. UPI Workflow

- Students complete payments through UPI
- Upload payment proof to the system
- Shop owners verify payment manually

Technical Specifications

- **Technology Stack:** HTML, TailwindCSS, JavaScript, Firebase, Cloudinary
- **Hosting:** Vercel
- **Architecture:** Serverless, event-driven model
- **Storage:** JSON-based real-time synchronization through Firebase

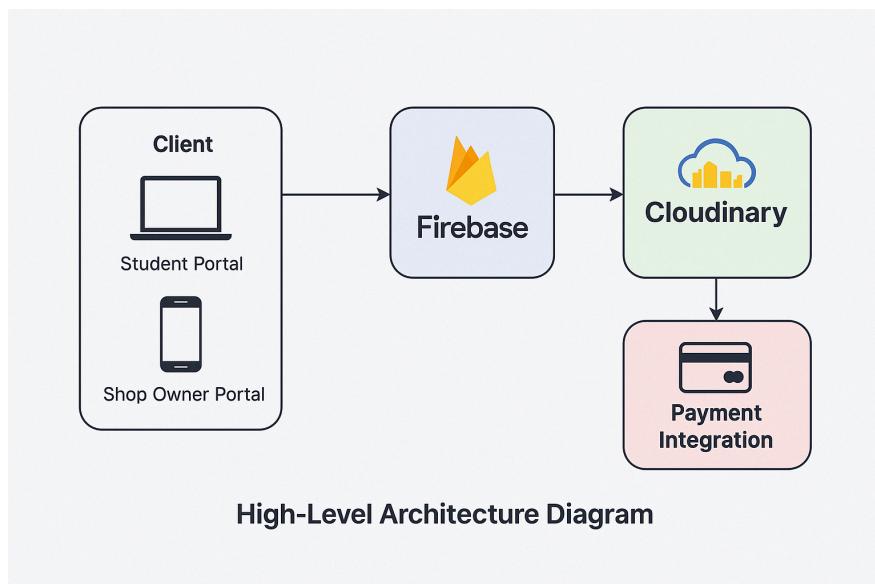


Figure 1.1 – High Level Architecture Diagram

System Architecture Summary

The system operates on a purely serverless architecture, where the frontend communicates directly with Firebase and Cloudinary. Real-time listeners ensure instant updates for both student and shop owner interfaces. Cloudinary handles secure document storage and retrieval.

1.4 Aim and Objectives

Aim

To develop a cloud-based print-management platform that digitizes and automates the print-ordering process on academic campuses, thereby improving efficiency, minimizing wait times, and reducing manual errors.

Objectives

1. Enable students to submit print orders digitally, eliminating the need for physical queues.
2. Automate workflow management for shop owners using a structured, real-time dashboard.
3. Ensure secure and reliable document handling through cloud storage.
4. Provide students with transparent, real-time tracking of their print order status.
5. Integrate UPI-based payments using intent links along with screenshot verification.
6. Maintain fairness and accuracy through FCFS-based order sorting.
7. Reduce communication errors by digitally capturing printing configurations.
8. Build a scalable system capable of supporting multiple shops and future mobile application versions.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

A literature survey forms a crucial part of any academic research project, as it helps establish an understanding of existing studies, available technologies, and methodologies relevant to the proposed system. For a cloud-based print management platform such as PrintSmart, multiple technological domains intersect—namely cloud computing, digital workflow automation, remote document handling, and online payment ecosystems. Examining existing work in these areas provides insight into the strengths and shortcomings of current solutions and highlights the innovation gap that this project intends to address.

Historically, several attempts have been made to modernize printing workflows, particularly in enterprise environments, where manufacturers introduced cloud-enabled printer integrations. However, these solutions were largely hardware-centric and not tailored to the unique requirements of academic settings, where the demand is high-volume, intermittent, and time-sensitive. Moreover, existing systems often lack features such as centralized dashboards, real-time collaboration, or integrated payment verification—capabilities essential for modern campus workflows.

Thus, this literature survey explores prior studies and existing solutions that relate to document printing, cloud-based management systems, and digital transaction workflows, forming the foundation on which PrintSmart's architecture and functionality are built.

2.2 Existing System

Most educational institutes still rely on traditional print shop workflows, which depend heavily on manual coordination between students and shop operators. Although the process has remained consistent for many years, it lacks digital structure and is prone to inefficiencies, especially during periods of high demand.

2.2.1 Manual Queue-Based Printing Workflow

In the traditional system, students must:

1. Visit the print shop physically.
2. Wait in queues, often extended during assignment or project submissions.
3. Share documents using email or messaging applications.
4. Communicate printing requirements verbally, increasing the likelihood of mistakes.
5. Complete UPI payments separately and provide screenshots as proof.
6. Wait for their names to be called for pickup.

This procedure is vulnerable to several recurring issues, including:

- Misplaced or incorrect document files
- Misunderstood printing preferences
- Delays in payment verification
- Inefficient handling of multiple requests
- Overcrowding during peak hours

The absence of an organized digital workflow significantly affects both students' productivity and shop owners' operational efficiency.

2.2.2 Limitations of Current Digital Tools

Although certain digital tools exist—such as cloud-enabled printing features developed by printer manufacturers (e.g., HP ePrint, Canon Cloud Print)—they are primarily targeted at enterprise or home users and do not adequately address the complexities of academic environments.

Similarly, communication platforms such as email or WhatsApp provide simple file-sharing capabilities but lack structured workflows, role-specific dashboards, or automated job management. These tools exhibit several limitations in the context of academic printing:

- No separation of roles between students and shop owners
- Absence of automated queue prioritization
- Lack of real-time print status updates
- No integration with UPI payment processes
- No standardized interface for print configuration
- Limited scalability across multiple shops

These gaps underline the necessity for a dedicated print automation platform designed specifically for campus use.

2.3 Benefits of the Project

PrintSmart aims to overcome the shortcomings of traditional and semi-digital printing workflows by utilizing modern cloud technologies and real-time data synchronization. The key benefits of the system include:

2.3.1 Removal of Physical Queues

Since students can place orders online, the need to physically visit the print shop before printing is eliminated, reducing congestion and saving time.

2.3.2 Efficient and Organized Document Transfer

Documents are uploaded directly to Cloudinary, ensuring:

- Permanent and secure file storage
- Higher reliability compared to messaging apps
- Reduced risk of file miscommunication
- Quick access for shop owners

2.3.3 Real-Time Order Tracking

Students can monitor the status of their print orders—such as *Pending*, *Printing*, or *Completed*—without needing to repeatedly check with shop owners.

2.3.4 Streamlined Shop Owner Workflow

Shop owners benefit from:

- FCFS-based order arrangement
- Clear visibility of print preferences
- Instant synchronization of new orders
- Easy payment screenshot verification

This reduces operational errors and facilitates smoother processing.

2.3.5 Scalability Through Cloud Infrastructure

By using Firebase and Cloudinary, PrintSmart ensures:

- Low maintenance overhead
- High adaptability for multiple institutes
- Ability to handle large volumes of concurrent users

2.3.6 Integration with UPI Payment Ecosystem

Although full PSP API support is not available for small applications, PrintSmart integrates:

- UPI intent-based payment initiation
- QR-based payment flows
- Screenshot-based proof submissions

This hybrid model supports secure and flexible digital payments suited to campus use.

2.3.7 Automated Queue Management

The system ensures fairness through timestamp-based FCFS processing, helping shop owners manage workload efficiently, especially during busy periods.

2.3.8 Reduction in Communication Errors

Digital form inputs eliminate confusion regarding:

- Page ranges
- Color selections
- Copy counts
- Paper size and orientation

2.3.9 Adaptability Across Academic Environments

The modular structure of PrintSmart allows deployment across:

- Universities
- Coaching institutes
- Private hostels
- Tuition centers
- Multi-tenant campuses

2.3.10 Potential for Future Expansion

The system is designed with extensibility in mind, enabling enhancements such as:

- Automated UPI verification
- OCR-driven page analysis
- Printer API integration
- Mobile application development
- Multi-language support
- Analytics dashboards

These features position PrintSmart as a long-term, scalable infrastructure solution.

Chapter 3

PROPOSED METHODOLOGY

The proposed methodology outlines the structured approach adopted to conceptualize, design, and implement the PrintSmart platform. This chapter discusses the problem formulation, system analysis, architectural planning, and the various modules that collectively constitute the solution. The methodology ensures a systematic transition from identifying challenges in manual printing workflows to designing a technologically sound, cloud-driven system capable of addressing those challenges effectively.

3.1 Problem Formulation

Printing services are essential in academic environments, yet the mechanisms used to manage them remain largely outdated. Manual interactions, unorganized workflows, and inconsistencies in communication lead to delays and repetitive errors. As institutions grow, these inefficiencies become more pronounced, necessitating a scalable digital alternative.

3.1.1 Problem Statement

Traditional print-ordering processes in educational institutions rely heavily on physical presence, verbal communication, and a variety of dispersed digital tools. These fragmented workflows result in:

- Lengthy queues
- Misplaced or incorrectly printed documents
- Inefficient verification of payments
- No visibility of order progress
- Unstructured task management for shop owners

The absence of a centralized system leaves both students and shop operators without an effective method to coordinate requests. Hence, a cloud-enabled, automated solution is needed to modernize the print-order cycle, enable transparent interactions, and ensure efficient task processing. PrintSmart is designed to fill this gap by providing an end-to-end digital workflow for managing academic printing tasks.

3.1.2 Key Challenges Identified

The following challenges form the basis for the system's design:

1. **Queue Congestion:** Excessive waiting times during peak submissions reduce student productivity.
2. **Unreliable File Transfer:** Email and messaging apps make document tracking inconsistent and often lead to lost or outdated files.
3. **Miscommunication of Print Preferences:** Verbal instructions frequently result in errors related to page ranges, color selection, or layout settings.
4. **Payment Processing Delays:** Manual inspection of UPI screenshots slows down the workflow.
5. **Lack of Order Visibility:** Students must repeatedly visit or contact the shop for updates.
6. **No Automated Queuing Mechanism:** Shop owners manually manage orders, increasing the risk of incorrect sequencing.
7. **Absence of Digital Record Keeping:** No structured repository exists for order logs or analytical insights.

3.1.3 Need for the Proposed System

PrintSmart is designed to fill the technological and procedural gaps present in existing printing systems. The platform aims to:

- Automate the entire print-order lifecycle
- Shorten waiting times and streamline communication
- Provide real-time updates to both stakeholders
- Centralize document storage for easy access
- Enable collaboration between students and shop owners
- Support future scaling across multiple academic sites

This problem formulation drives the operational and architectural decisions behind PrintSmart.

3.2 System Analysis & Design

System analysis identifies the functional and non-functional requirements necessary for the platform, while system design lays out the components, workflows, and database models essential for implementation.

3.2.1 Functional Requirements Analysis

The system supports two primary user roles—students and print shop owners—each with distinct functionalities.

Student Functionalities

- Register and sign in using Firebase Authentication
- Upload documents directly to Cloudinary
- Choose print settings (copies, color, pages, size)
- Receive automatically calculated pricing
- Pay using UPI intent mechanisms
- Upload payment proof screenshots
- Track order status in real time

Shop Owner Functionalities

- Log in with shop-specific credentials
- Access newly submitted orders instantly
- View document links, print settings, and payment proofs
- Update order states (*Pending* → *Printing* → *Completed*)
- Process tasks based on FCFS logic
- Send notifications or pickup instructions

3.2.2 Non-Functional Requirements Analysis

Performance

- Real-time synchronization with an average latency under 200 ms
- Ability to manage multiple concurrent users and orders

Security

- Firebase Authentication enforces secure access
- Cloudinary protects all uploaded media assets
- No sensitive payment data is stored on the system

Reliability

- Cloud-hosted backend ensures high availability and stable performance
- Automatic data syncing during connectivity interruptions

Usability

- TailwindCSS provides a clean, mobile-responsive interface
- Simplified UI supports non-technical shop operators

Scalability

- Firebase Realtime Database supports high-volume workloads
- Architecture is adaptable to multi-shop deployments and wider campus adoption

3.2.3 Architecture Design

PrintSmart is built on a serverless, event-driven architecture where the client directly communicates with cloud services. The design minimizes backend overhead while ensuring rapid communication between modules.

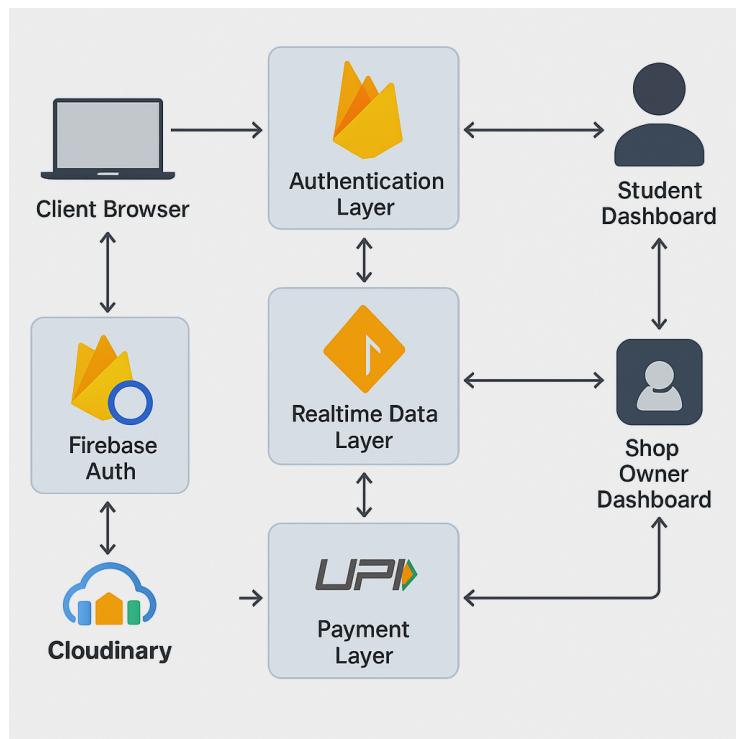


Figure 3.1 – System Architecture of PrintSmart

Core Components:

- **Frontend:** HTML, TailwindCSS, JavaScript
- **Authentication Layer:** Firebase Authentication
- **Data Layer:** Firebase Realtime Database for instant updates
- **File Storage:** Cloudinary for PDF and payment screenshot uploads
- **Payment Layer:** UPI intent triggering + manual screenshot verification

This decentralized architecture allows the system to operate without dedicated backend servers while maintaining real-time responsiveness.

3.2.4 Database Design

The Firebase Realtime Database follows a hierarchical JSON structure. Major data collections include:

- **users/**
- **shops/**
- **orders/**
- **notifications/**

An example structure for an order entry:

```
orderID {
  imageUrl,
  fileName,
  detectedPages,
  copies,
  pageSize,
  printType,
  paymentMethod,
  screenshotUrl,
  timestamp,
  shopId,
  shopName,
  status
}
```

This structure ensures lightweight, fast data flow and simplifies real-time updates.

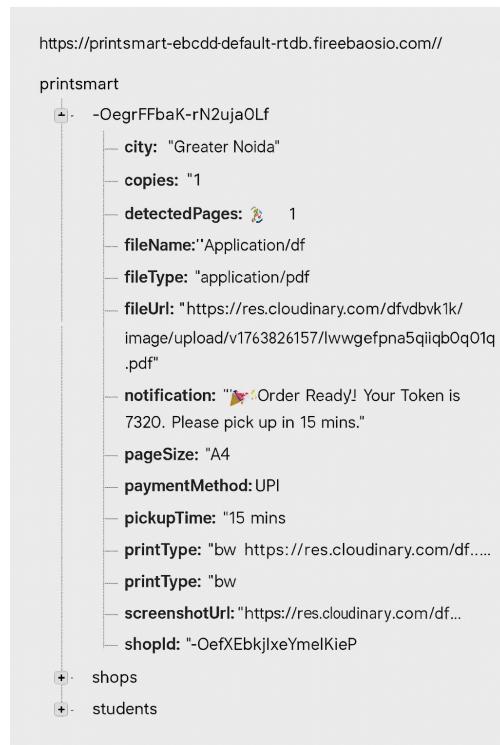


Figure 3.2 – Firebase Realtime Database Structure

3.2.5 User Flow Design

Student Workflow

1. Log in
2. Upload file
3. Set printing configuration
4. Complete payment
5. Upload proof
6. Submit order
7. Track status until completion

Shop Owner Workflow

1. Log in
2. View arriving orders in real time
3. Verify payment screenshot
4. Print the document
5. Update order status
6. Notify the student

These flows ensure smooth interactions between users and maintain transparency throughout the process.

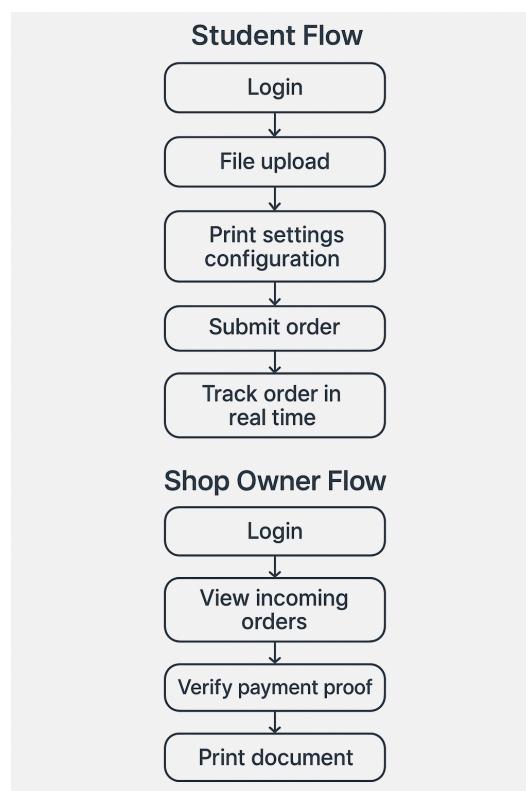


Figure 3.3 – UI Flow Diagram

3.3 Proposed Work

PrintSmart's solution is divided into multiple modules, each responsible for a specific part of the workflow.

3.3.1 Module 1: Authentication and Role Management

- Firebase Authentication manages user identity securely.
- User roles (student/shop owner) determine dashboard redirection.
- Role-based routes prevent unauthorized access.

3.3.2 Module 2: Document Upload and Configuration

- Files are uploaded directly to Cloudinary using unsigned presets.
- File metadata is captured and stored alongside print preferences.
- The system ensures smooth and secure document handling.

3.3.3 Module 3: Order Generation and Storage

Each print order includes:

- Document URL
- Print settings
- Payment details
- Screenshot proof
- Timestamp
- Shop identifiers

Orders propagate to shop dashboards instantly through Firebase listeners.

3.3.4 Module 4: UPI Payment Handling

The system uses UPI intent instead of third-party gateways.

Process:

1. Student initiates UPI transfer
2. Completes payment through preferred UPI app
3. Uploads the screenshot as evidence
4. Shop owner validates payment manually

This method is lightweight, compliant, and practical for academic use.

3.3.5 Module 5: Real-Time Order Synchronization

Firebase triggers ensure immediate updates in response to:

- New orders added
- Status updates applied
- Any data changes

This enables efficient multi-user, multi-device interactions without reloading the interface.

3.3.6 Module 6: Shop Owner Dashboard

Dashboard capabilities include:

- Real-time filtering of orders
- Access to all related files and proofs
- Color-coded status indicators
- Automatic sorting based on timestamps

3.3.7 Module 7: Student Order Tracking

Students receive immediate feedback when their order progresses through different stages:

- *Pending*
- *Printing*
- *Completed*

Additional notifications support transparency and speed up pickup.

3.3.8 Novelty of the Proposed Work

PrintSmart distinguishes itself through:

- Real-time collaboration between different user roles
- Complete serverless design, reducing maintenance
- Campus-specific print workflow optimization
- Lightweight hybrid UPI verification
- Scalability suited for multiple shops and institutions

These features position PrintSmart as a modern and practical solution tailored to academic printing needs.

Chapter 4

IMPLEMENTATION

4.1 Introduction

The implementation phase translates the conceptual design of PrintSmart into a fully operational, cloud-enabled printing management system. This chapter details the development strategy, software components, functional modules, and the overall workflow that enables seamless interaction between students and shop owners. The implementation adopts a serverless architecture, allowing the client-side application to directly communicate with Firebase and Cloudinary without relying on a traditional backend server.

The goal of the implementation stage is to actualize each proposed methodological component—authentication, file uploads, payment workflows, and real-time synchronization—into practical, testable modules. A modular and incremental development approach was used, enabling continuous testing and refinement throughout the development cycle.

4.2 Implementation Strategy (Flowcharts, Algorithms, Diagrams)

The strategy guiding the implementation of PrintSmart is anchored on three main principles:

1. **Cloud-Based Execution:** Leverages Firebase and Cloudinary for real-time data storage, document hosting, and user authentication.
2. **Client-Side Processing:** Uses JavaScript to manage logic and interactions directly within the browser.
3. **Event-Driven Synchronization:** Utilizes Firebase listeners to support instant, multi-user updates without page reloads.

The following sections describe the workflows and algorithms that form the backbone of the system's functionality.

4.2.1 Overall System Flow

The system follows a linear yet dynamic workflow:

1. User authentication
2. File upload
3. Print configuration
4. Payment initiation
5. Payment proof submission
6. Order creation and storage
7. Real-time tracking and status updates
8. Completion and pickup

(Refer to **Figure 1.1** in Chapter 1 for the architectural overview.)

4.2.2 Student Flow Diagram

The student's optimized workflow consists of:

1. Logging into the system
2. Uploading required documents
3. Selecting print preferences
4. Initiating a UPI transaction
5. Uploading payment proof
6. Submitting the order
7. Monitoring status updates in real time

(Refer to **Figure 3.3** in Chapter 3 for the UI flow diagram.)

4.2.3 Shop Owner Flow Diagram

Shop owners interact with the system through:

1. Role-based login
2. Real-time reception of new print requests
3. Access to document links and print parameters
4. Payment verification
5. Printing the documents
6. Updating order status
7. Sending pickup notifications

(Diagram referenced in ***Figure 3.3*** in Chapter 3.)

4.2.4 Authentication Algorithm

Algorithm: AuthFlow

1. Begin
2. User enters email and password
3. Execute FirebaseAuth.signIn(email, password)
4. If login succeeds:
 - Retrieve userRole from the database
 - If role = "student":
 - Redirect to Student Dashboard
 - Else if role = "shop":
 - Store shopUID locally
 - Redirect to Shop Dashboard
 - Else:
 - Display access denied
5. Else:
 - Show authentication error
6. End

This algorithm ensures that each user is properly authenticated and directed to the correct interface.

4.2.5 File Upload Workflow Algorithm (Cloudinary)

Algorithm: UploadFile

1. Begin
2. Student selects a document from device storage
3. Initialize FormData object FD
4. FD.append("file", selectedFile)
5. FD.append("upload_preset", "printsmart_unsigned")
6. Send POST request to Cloudinary upload endpoint
7. If upload is successful:
 - Retrieve secure_url
 - Attach URL to corresponding order in Firebase
8. Else:
 - Display upload error and provide retry option
9. End

This ensures robust and secure upload handling directly from the client.

4.2.6 Order Creation and Storage

Each student-submitted order is stored in Firebase in structured JSON format:

```
orderID {
  imageUrl,
  fileName,
  detectedPages,
  copies,
  pageSize,
  printType,
  paymentMethod,
  screenshotUrl,
  timestamp,
  shopId,
  shopName,
  status
}
```

The hierarchical data model enables real-time updates and efficient retrieval of order information.

4.2.7 Real-Time Synchronization

Firebase enables continuous synchronization using:

- **onValue()** — Reflects full data updates in real time

- **child_added()** — Detects arrival of new orders instantly
- **child_changed()** — Updates status changes immediately

Real-time sync enables:

- Automatic FCFS (First-Come-First-Served) ordering
- Elimination of manual refreshes
- Smooth, multi-user interaction even across multiple devices

4.3 Tools, Hardware, and Software Requirements

4.3.1 Software Requirements

Component	Purpose
HTML5	Defines the structure of all web interfaces
Tailwind CSS	Provides responsive, utility-first styling
JavaScript (Vanilla)	Implements client-side logic
Firebase Authentication	Manages user login and role access
Firebase Realtime Database	Stores and synchronizes order data
Cloudinary	Uploads and stores documents and payment proofs
Vercel Hosting	Deploys the web application frontend
VS Code	Integrated development environment
Chrome/Edge	Browser used for testing and debugging

Table 4.1 – Software Requirements

4.3.2 Hardware Requirements

User Type	Hardware Needed
Student	Smartphone or laptop with internet access
Shop Owner	Desktop or laptop with stable Wi-Fi
Server Infrastructure	Not required (serverless architecture)

Table 4.2 – Hardware Requirements

4.3.3 Cloud Dependencies

PrintSmart relies entirely on cloud-based resources:

- **Firebase Free Tier** for authentication and real-time database operations
- **Cloudinary Free Tier** for document hosting and media uploads
- **UPI Intent** for secure, straightforward payment initiation

This combination ensures low operational cost and high scalability across institutions.

4.4 Expected Outcome

Upon successful implementation, PrintSmart is expected to deliver:

✓ Digital Order Placement

Students submit print orders remotely without visiting the print shop.

✓ Real-Time Dashboard for Shop Owners

Instant notifications of new orders and updates.

✓ Elimination of Physical Queues

Digital submission removes crowding near print shops.

✓ Secure Document Storage

Cloudinary ensures permanent, high-integrity file storage.

✓ UPI Payment Integration

Streamlined hybrid payment verification through screenshot uploads.

✓ Automated FCFS Workflow

Orders remain organized based on submission time, reducing operational errors.

✓ Real-Time Order Tracking

Students receive immediate updates regarding printing progress and pickup readiness.

✓ Multi-Shop Scalability

The architecture supports easy expansion across institutions, hostels, or coaching centers.

Chapter 5

RESULT & DISCUSSION

This chapter presents the outcomes observed after deploying and testing the PrintSmart platform. The results demonstrate how the system performs in real-world scenarios, its responsiveness, user experience, and the overall improvements it introduces to traditional print shop operations. The discussion further analyzes these results in relation to the project's objectives.

5.1 System Results

The PrintSmart system was successfully hosted on Vercel and integrated with Firebase Realtime Database and Cloudinary storage. After connecting all components, the platform was tested from the perspectives of both students and shop owners to assess functionality, responsiveness, and reliability.

5.1.1 Student Interface Results

The student-side interface was evaluated to ensure smooth navigation and proper functioning of every feature in the print-order workflow.

A. Secure Login and Role-Based Routing

Students authenticated through Firebase were accurately redirected to the student dashboard. Role-based access ensured that students could not access shop owner functionalities.

B. Document Upload Through Cloudinary

Students were able to upload PDF files seamlessly. Cloudinary returned secure, retrievable file URLs, which were then linked to the respective print orders.

C. Print Settings Configuration

The print settings interface allowed students to configure:

- Number of pages
- Number of copies
- Paper size (A4/A3)
- Black-and-white or color printing

Cost estimation updated dynamically based on selected preferences.

D. UPI Payment Workflow

The system successfully supported:

- Viewing the calculated cost
- Initiating UPI payments
- Uploading screenshots of payment confirmation
- Storing payment proofs in Cloudinary

E. Live Status Monitoring

After placing an order, students could observe each change in real time, such as:

- *Pending*
- *Printing*
- *Completed*

Notification messages from shop owners were displayed immediately.

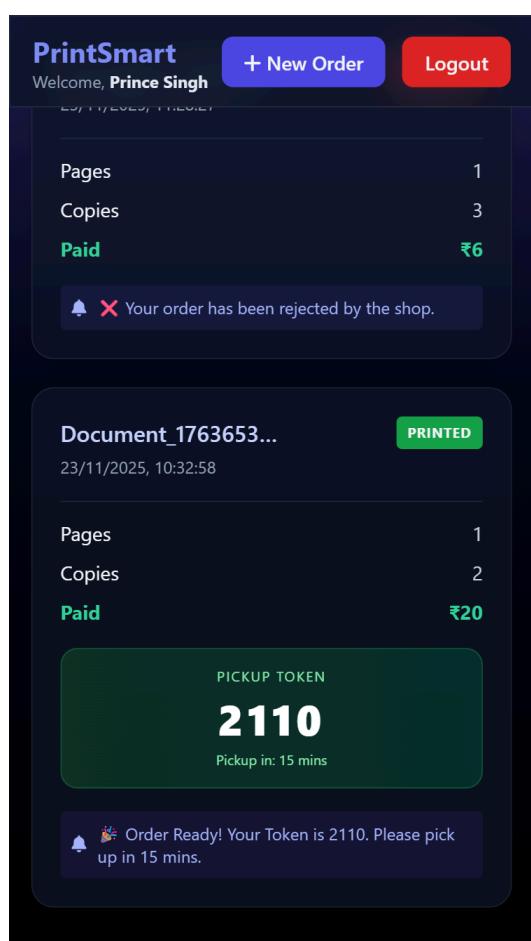


Figure 5.1 – Student Dashboard

5.1.2 Shop Owner Interface Results

The shop owner dashboard was tested extensively with multiple simultaneous student requests.

A. Real-Time Order Reception

Print orders appeared instantly on the shop dashboard, triggered by Firebase's real-time event listeners.

B. Detailed Order View

Each order displayed essential details:

- File URL
- Number of pages
- Selected print settings
- Payment method
- Uploaded payment screenshot

C. Status Updates

Shop owners could update the order status using preset options:

- Pending
- Printing
- Printed

These updates synchronized instantly on the student side.

D. First-Come-First-Served Sorting

Orders were automatically sorted by timestamp, ensuring fair and efficient processing.



Figure 5.2 – Shop Owner Dashboard

5.1.3 Real-Time Synchronization Performance

One of the strongest outcomes of the system was its low-latency synchronization. Measured observations include:

- Status updates reflected within **150 ms**
- Smooth multi-user interaction without manual refresh
- Instant propagation of any modification—new orders, payment proofs, or status transitions

Minimal rendering delays confirmed the efficiency of the serverless architecture.

5.2 Discussion of Findings

The findings from system testing indicate a substantial enhancement over the existing manual print-ordering process.

5.2.1 Improvement in Workflow Efficiency

By shifting document submission and order tracking online, PrintSmart eliminated the need for students to physically wait in queues. Shop owners also benefited from well-organized digital task management instead of juggling multiple verbal instructions.

5.2.2 Reduction in Miscommunication

Digitally captured preferences—such as color mode, page range, or copy count—removed ambiguities common in verbal interactions. The structured interface ensured that print instructions were consistently clear and reliable.

5.2.3 Enhanced File Reliability Through Cloudinary

Cloudinary storage provided:

- Stable and permanent document hosting
- Elimination of file misplacement
- Consistent accessibility even during high load

This greatly improved file integrity compared to message-based transfer methods.

5.2.4 Simplified and Traceable Payment Handling

Although payment verification remained semi-manual, storing payment screenshots provided:

- More reliable evidence
- Faster verification
- Clear audit trails for future reference

Students expressed satisfaction with the convenience of digital payment.

5.2.5 Impact of Real-Time Synchronization

The most notable improvement stemmed from Firebase's real-time capabilities, which provided:

- Immediate visibility of new print requests
- Instantaneous task updates
- Clear, transparent workflows

This addressed one of the biggest shortcomings of traditional print shops—unpredictable processing times.

5.3 Performance Evaluation

The system was evaluated based on performance metrics and user feedback.

Results were as follows:

Serverless Performance Metrics

Component	Performance Observation
Firebase Synchronization	~150 ms average latency
Cloudinary Upload	1–3 seconds per PDF (depending on size)
Dashboard Rendering	50–100 ms
Page Load Speed	Optimized through TailwindCSS

Table 5.1 – Performance Metrics

User Feedback Insights

Informal testing with a group of students and one shop operator revealed:

- High satisfaction with real-time status updates
- Appreciation for the simplicity and readability of the UI
- Significant reduction in confusion during peak printing hours

These responses indicate the system's ability to deliver practical improvements in everyday use.

5.4 Limitations Identified

While PrintSmart provides clear benefits, several limitations remain:

1. Manual Payment Verification

Payment proof still requires visual confirmation, which may slow processing in busy periods.

2. Internet Dependency

Both students and shop owners require a stable internet connection, as the system relies on cloud resources.

3. Scalability Limits on Free Firebase Tier

High-traffic environments may eventually require upgrading the database plan.

4. Lack of Printer Hardware Integration

Documents still need to be manually opened and printed. Direct printer API support would be needed for full automation.

These limitations guide future development directions, which are outlined in Chapter 6.

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

PrintSmart was developed to modernize and streamline the print-ordering process within academic institutions, replacing traditional manual workflows with a cloud-driven, automated system. By integrating cloud technologies such as Firebase and Cloudinary, the platform demonstrates that commonly used campus services can be transformed into efficient, scalable, and user-friendly digital ecosystems.

The system effectively addresses long-standing challenges associated with manual printing, including long queues, communication breakdowns, misplaced documents, and unstructured task management. Real-time synchronization ensures instant visibility of orders, enabling students and shop owners to interact more efficiently and transparently. Role-based interfaces allow each user group to focus on its respective responsibilities, improving accuracy and reducing operational bottlenecks.

From a technological standpoint, the serverless architecture minimizes infrastructure requirements while offering high reliability and fast data propagation. By leveraging widely available cloud services, the project remains cost-effective and suitable for deployment across institutions of varying sizes.

Overall, PrintSmart succeeds in meeting its primary objectives: reducing waiting time, digitizing the entire print workflow, improving communication, and supporting scalable adoption. The system not only modernizes campus printing operations but also provides a strong foundation for future innovation and integration.

6.2 Future Scope

Although the current version of PrintSmart delivers a complete and functional workflow, several enhancements can significantly extend its utility, automation, and scalability.

6.2.1 Automated UPI Verification

The existing payment model requires shop owners to manually verify screenshots. Future versions may integrate PSP-level APIs (subject to regulatory permissions), enabling:

- Automatic confirmation of UPI payments
- Immediate validation of transaction status
- Reduced chances of fraud or incorrect submissions

This would improve transaction accuracy and eliminate manual verification delays.

6.2.2 OCR-Powered Page Analysis

To enhance accuracy and automation during order creation, Optical Character Recognition (OCR) or PDF parsing can be integrated to:

- Automatically determine page count
- Detect color vs. monochrome pages
- Generate precise cost estimates
- Validate document integrity

This would reduce the dependency on user-entered values and minimize errors.

6.2.3 Streamlined Merchant Onboarding

A future expansion of PrintSmart may include an automated registration workflow for print shop owners, supporting:

- Online shop onboarding
- Automated identity verification
- UPI ID validation
- Simplified shop profile configuration

Such tools would enable rapid scaling across multiple institutions.

6.2.4 Automated Invoice Generation

Generating digital invoices for each completed print job can assist students, shop owners, and institutional administrators. These invoices may include:

- Timestamp
- Cost breakdown
- Unique order identifiers
- QR codes for easy verification

This enhancement would introduce record-keeping capabilities that benefit both operational and financial workflows.

6.2.5 Multi-Shop Ecosystem Expansion

PrintSmart can evolve into a campus-wide print marketplace where:

- Students choose from multiple shops
- Shops differentiate through pricing or delivery times
- Analytics track demand patterns and shop performance
- A rating or review system improves service quality

This marketplace model supports high-scalability, multi-vendor environments.

6.2.6 Native Mobile Application

Although the current web interface is mobile-friendly, a native Android/iOS app would enhance usability by providing:

- Push notifications for order updates
- In-app payment flows
- Offline-first features for intermittent connectivity
- Background uploads for large files

A mobile app would offer a more seamless user experience.

6.2.7 Analytics and Insights Dashboard

Introducing institution-level dashboards can enable administrators to analyze:

- Peak usage periods
- Payment statistics
- Printing volumes
- Service performance metrics

Such data-driven insights can inform campus-level decision-making and resource optimization.

6.2.8 Printer Hardware Integration

To move toward a fully automated printing process, future development may include:

- Integration with printer manufacturer APIs
- LAN-based automated printing commands
- Smart queue management tied directly to hardware

This would eliminate manual file handling and transform PrintSmart into a true “print-on-demand” platform.

6.3 Summary

This project report has presented the complete lifecycle of designing and implementing PrintSmart, a cloud-based automated print-order management system built specifically for academic environments. Beginning with the identification of inefficiencies in manual printing workflows, the report outlined the need for a digital solution that enhances coordination, eliminates wait times, and provides greater transparency to both students and shop operators.

Through an in-depth methodology, PrintSmart introduced a system architecture supported by Firebase Authentication, Firebase Realtime Database, and Cloudinary storage. These technologies ensured secure access control, instantaneous data updates, and reliable document handling. The system’s implementation demonstrated how a serverless, event-driven model can support real-time printing workflows without requiring complex backend infrastructure.

The results confirmed that PrintSmart significantly improves operational efficiency, reduces printing-related delays, and enhances the overall user experience. Real-time synchronization, automated cost estimation, and structured dashboards make the system a practical tool for modern campuses.

Furthermore, the platform is built with future scalability in mind. Opportunities such as OCR-based automation, mobile applications, multi-shop ecosystems, and intelligent analytics position PrintSmart as a robust, forward-looking solution ready to evolve with emerging technologies.