ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

E-Receipt Generator Using Cloud Computing

Swathi A.[1]; Naveen Kumar B.[2]

[1]PG, Research Scholar, DoS in CSE University BDT College of Engineering Davanagere, Karnataka, 572105, India
[2]Assistant Professor, DoS in CSE, University BDT College of Engineering Davanagere, Karnataka, 577004

Publication Date: 2025/11/12

Abstract: Traditional paper-based receipt systems are increasingly seen as environmentally unsustainable, operationally inefficient, and vulnerable to fraud. With growing digital transformation, cloud computing offers scalable, secure, and eco-friendly alternatives. This paper presents the design and implementation of an E- Receipt Generator using Cloud Computing, developed with Flask (Python) for the backend, Bootstrap for the frontend, AWS RDS (MySQL) for cloud storage, and Chart.js for analytics. The system enables dynamic item entry, automatic calculation of totals and GST, UPI QR code generation for seamless payments, and retrieval of receipts with visualization dashboards for business insights. Results demonstrate improved transaction efficiency, enhanced security, and significant reduction in paper usage, positioning this solution as a sustainable alternative to conventional receipts. Future directions include block chain integration for immutable storage and AI-driven sales analytics.

Keywords: *Digital Receipts, Cloud Computing, Sustainable IT, Flask, AWS RDS, QR Code Payments, Analytics Dashboard.*

How to Cite: Swathi A.; Naveen Kumar B. (2025) E-Receipt Generator Using Cloud Computing. *International Journal of Innovative Science and Research Technology*, 10(11), 250-256. https://doi.org/10.38124/ijisrt/25nov181

I. INTRODUCTION

Cloud computing has emerged as a transformative paradigm, providing on-demand, scalable, and cost- efficient IT services. Industries are increasingly leveraging cloud platforms to modernize legacy systems, optimize operations, and enable digital sustainability. In parallel, traditional receipt management systems face growing scrutiny due to their environmental impact, inefficiency, and susceptibility to loss and fraud. Paper receipts, though widely used, contribute significantly to environmental degradation. Millions of trees are cut down for paper production annually, and most receipts are discarded within hours of issuance. From a business perspective, paper receipts create challenges in recordkeeping, warranty management, and sales analytics. Customers often lose receipts, while businesses incur additional costs in printing and storage. This project introduces an E-Receipt Generator using Cloud Computing, offering a secure, scalable, and eco- friendly alternative to traditional receipts. Unlike existing systems that focus solely on digitization, this solution integrates UPI QR payments, real-time analytics dashboards, and scalable cloud storage to address modern business needs.

➢ *Contributions of the Work:*

- Proposes a sustainable receipt management system powered by cloud computing.
- Implements real-time UPI QR code generation for secure, seamless transactions.
- Provides dynamic, multi-item entry with automated calculation of totals.
- Integrates an analytics dashboard for business intelligence using Chart.js.
- Demonstrates a cloud deployment pipeline with AWS EC2 and RDS.

- *Global Trends in Digital Transformation*

The shift from physical to digital systems has become essential for competitiveness, accelerated by COVID-19. E-receipts offer a safe, sustainable alternative, reducing printing and storage costs by up to 40% while improving accessibility.

- *The Sustainability Imperative*

Over 300 billion paper receipts are printed annually, consuming vast natural resources and often containing harmful chemicals like BPA. E-receipts align with UN SDGs, supporting responsible consumption and climate action.

- *Technological Gaps in Existing Systems*

Many SMEs rely on outdated receipt methods lacking cloud integration, analytics, and digital payment support. Customers also struggle with receipt retrieval and expense tracking, highlighting the need for a secure, cloud-based solution.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

- Role of Cloud Computing in Digital Receipts Cloud platforms like AWS RDS enable scalable, reliable storage with backups and real-time access.

Combined with lightweight frameworks like Flask and Bootstrap, they ensure accessibility across devices.

- Integration with Digital Payments and Analytics Growing adoption of UPI and QR payments demands seamless integration with receipts. Embedding analytics allows businesses to monitor sales, forecast trends, and make data-driven decisions, setting this system apart from traditional models.

- *Research Motivation & Objectives*

This study aims to develop a sustainable e-receipt system that:

✓ Leverages cloud computing for scalability.
✓ Integrates UPI QR for digital payments.
✓ Provides analytics dashboards for insights.
✓ Ensures secure, accessible records for businesses and customers.

II. LITERATURE REVIEW

- Cloud Computing Foundations and Market- Oriented Models Cloud computing provides on-demand, elastic resources and a pay-as-you-go model that shifts CAPEX to OPEX, enabling rapid provisioning and global accessibility for applications such as e- receipt systems. Market-oriented cloud architectures (Buyya et al.) emphasize resource metering, SLA negotiation, and pricing mechanisms that enable multi-tenant services to scale economically. Several surveys highlight how cloud platforms enable SMEs to adopt enterprise- grade services without heavy infrastructure investment, making cloud the logical foundation for scalable receipt management and analytics platforms [Buyya et al., 2009; Saad et al., 2022]

- Digital receipt systems: architectures and use cases A growing body of research explores replacing thermal paper receipts with digital alternatives. Asriani and Arnous (2025) showcase a web-based system for toll payments, highlighting cost savings and reduced paper waste. Kumar et al. (2019) combine IoT devices with cloud storage for automated receipts but omit integrated analytics.

Chen, Zhang, and Liu (2020) propose a blockchain–cloud model for tamper-proof receipts, though adoption is hindered by higher cost and complexity. Case studies across sectors consistently report benefits such as lower printing costs, easier retrieval, and better customer convenience. Approaches generally trade-off between simplicity and affordability (cloud DB + QR) versus stronger auditability (blockchain). The present system using AWS RDS for scalable storage with QR-based payments follows the former path, making it more practical for SMEs seeking low-cost, high-adoption solutions.

- Payment integration and QR/UPI ecosystems QR-based payments and UPI have rapidly become dominant channels for retail and micro-merchants due to their low customer friction and minimal infrastructure needs. Prior studies (Patel, 2022; Asriani & Arnous, 2025) highlight that QR/UPI integration reduces transaction errors and speeds checkout in smartphone-driven markets. From a systems perspective, effective integration links each receipt with its corresponding payment confirmation. This is achieved by generating a unique QR code tied to a receipt record and updating its status (paid, pending, failed) in the cloud database. Such a synchronous payment– receipt lifecycle, combined with asynchronous status monitoring, ensures reliability in real-world deployments.

- Blockchain and immutable receipt storage Several authors investigate blockchain to ensure receipt immutability and non-repudiation. Chen et al. (2020) outline architectures that store receipt hashes on-chain while keeping actual receipt payloads in cloud or off-chain storage, achieving tamper evidence with manageable on-chain costs.

Punia et al. (2024) survey blockchain-based access control, noting improved auditability but also higher complexity, performance overhead, and compliance challenges. Most literature recommends hybrid approaches cloud storage for performance and cost; blockchain for high-value or legally sensitive receipts requiring an immutable audit trail.

✓ Takeaway: Blockchain is attractive for auditability but may be overkill for routine retail receipts. Hybrid options (hash anchoring) can provide a compromise.

- Analytics, BI and decision support from receipts Digital receipts act as structured transaction data that enable business intelligence. Work by Kumar et al. and contemporary industry reports show analytics can deliver actionable insights: product- level sales shares, temporal demand patterns (daily/weekly/monthly), customer purchase frequency, and stock turnover. Charting libraries (Chart.js, D3) enable embedding interactive dashboards into web apps. Research also demonstrates that lightweight BI tools increase SME decision quality without large investments. Important point: For analytics to be actionable, the data model must capture structured item-level records (product codes, prices, quantities, timestamps), and the system must support aggregation queries. Cloud DBs (RDS) with proper indexing and periodic aggregation jobs meet this need.

- Security and privacy in cloud workflows Security and privacy are central concerns for cloud- hosted receipts. Soveizi et al. (2022) systematically review threats and mitigation techniques in cloud workflows, stressing encryption at rest/in transit, robust access control, identity management, and auditing. Receipt systems often contain

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

personal data (emails, phone numbers, partial card data), so GDPR-style privacy considerations, secure storage, minimal data retention, and consent mechanisms are essential. Storage security and data auditing protocols (e.g., remote attestation, logging, integrity checks) have been proposed to reduce tampering and support compliance. Implementation implications: Use TLS for all endpoints; store sensitive fields hashed/encrypted; implement role-based access for admin/cashier/customer; maintain tamper-evident logs.

- Deployment, scalability, and operational considerations Operational papers and case reports highlight key reliability and cost tradeoffs. AWS RDS (managed MySQL) offers automated backups, multi-AZ failover, and monitoring—important for systems that must retain transaction records long term. EC2 or containerized services (ECS/EKS) host the application layer; autoscaling handles peak loads. Several studies on cloud adoption (Saad et al., Milhem et al.) show SMEs prefer managed services because they offload infrastructure maintenance and reduce upfront costs. Energy efficiency and cost control (operational expenditure) are also recurring themes: cloud services enable elastic resource usage aligned with pay-as-you-go economics.

III. METHODOLOGY

This chapter describes the methodology adopted to design, develop, and deploy the E-Receipt Generator using Cloud Computing. The methodology follows an Agile development cycle, integrating incremental design, testing, and deployment. The system is engineered with modular components backend (Flask), frontend (Bootstrap + HTML/CSS/JS), cloud database (AWS RDS), analytics dashboard (Chart.js), and UPI QR-based payment integration. Together, these modules provide a secure, scalable, and eco-friendly solution for receipt generation and management.

➢ *Research Approach*

The project follows a practical system development methodology, combining Agile development practices with cloud-native architectural principles. Agile was chosen due to its adaptability, iterative feedback, and ability to refine features incrementally. The research approach involves:

- Requirement analysis based on limitations of traditional receipt systems.
- Iterative development of backend, frontend, and database modules.
- Integration of QR payment and analytics dashboard.
- Cloud deployment and performance testing.
- User evaluation and feedback collection.

➢ *Agile Development Model*

The project adopts the Agile Scrum framework, where development is divided into sprints, each focusing on specific modules (receipt creation, QR payment, analytics, cloud deployment).

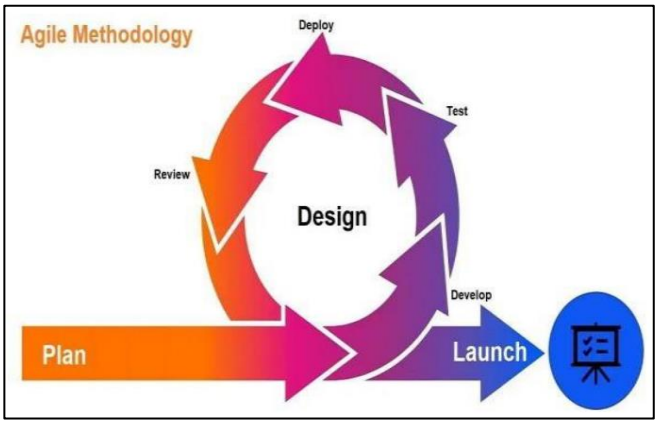

Fig 1 Agile Development Cycle

- Sprint 1: Requirement gathering and UI design. Sprint 2: Backend route development and database schema design.
- Sprint 3: Integration of QR code generation and payment workflows.
- Sprint 4: Dashboard and analytics module.
- Sprint 5: Deployment on AWS EC2 and testing with AWS RDS.

➢ *System Architecture*

The system follows a three-tier cloud-based architecture:

- *Presentation Layer (Frontend):*

✓ Built using HTML, CSS, Bootstrap.
✓ Provides input forms for receipt generation and displays analytics dashboards.

- *Application Layer (Backend):*

✓ Implemented in Flask (Python).
✓ Handles receipt creation, retrieval, payment integration, and analytics logic.

- *Data Layer (Cloud Database):*

✓ AWS RDS (MySQL) stores structured receipt and transaction data.
✓ Provides scalability, security, and automated backups.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

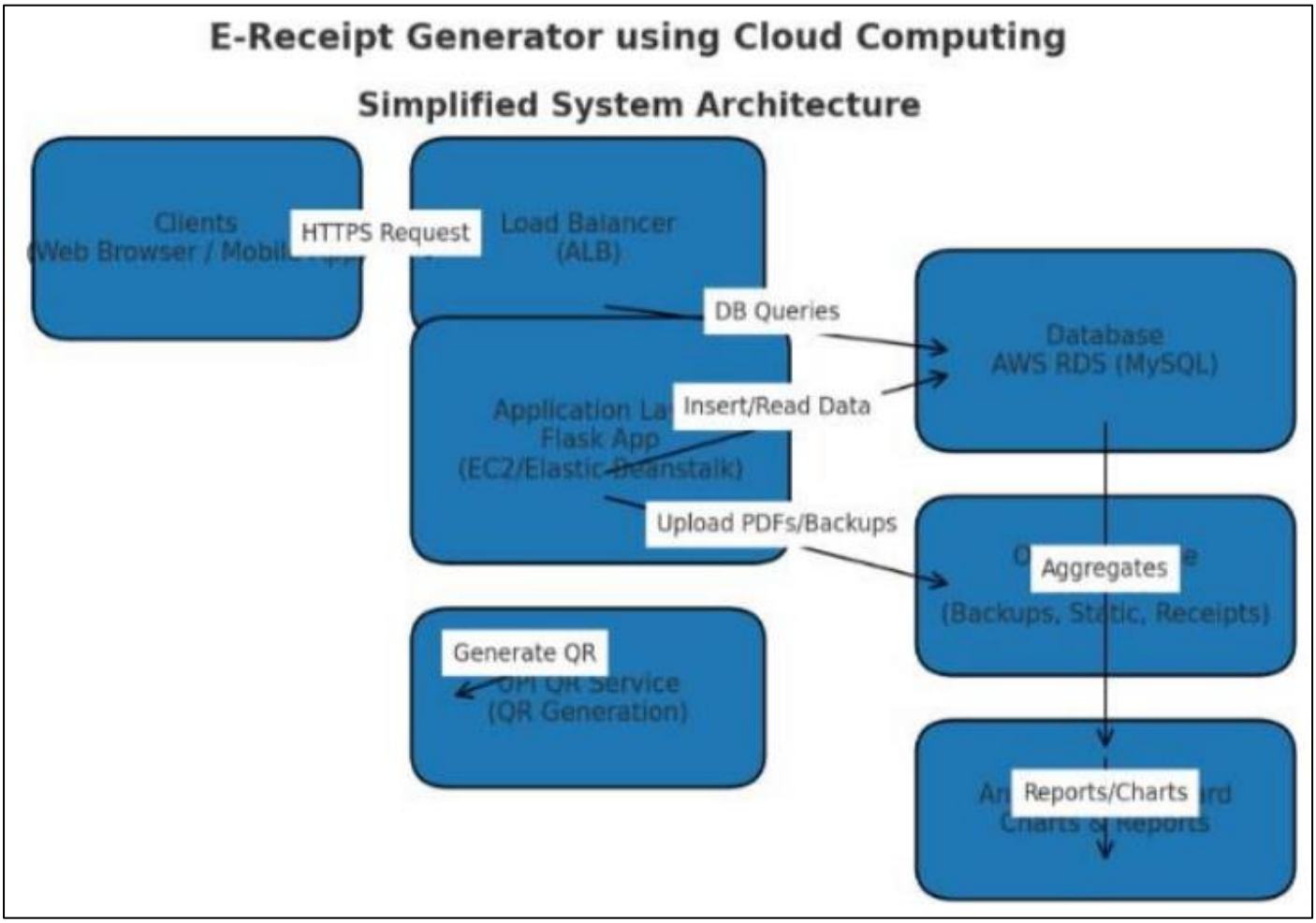

Fig 2 System Architecture Diagram

➢ *Workflow of the E-Receipt Generator*

The workflow illustrates the end-to-end receipt lifecycle.

- *Steps:*

✓ Customer purchases product(s).
✓ Merchant inputs items dynamically in frontend.
✓ Subtotal, GST, and grand total auto-calculated.
✓ Backend generates unique receipt ID and stores details in AWS RDS.
✓ UPI QR code dynamically generated for payment.
✓ Receipt generated digitally and made available for download/email.
✓ Analytics dashboard updates with sales data in real time.

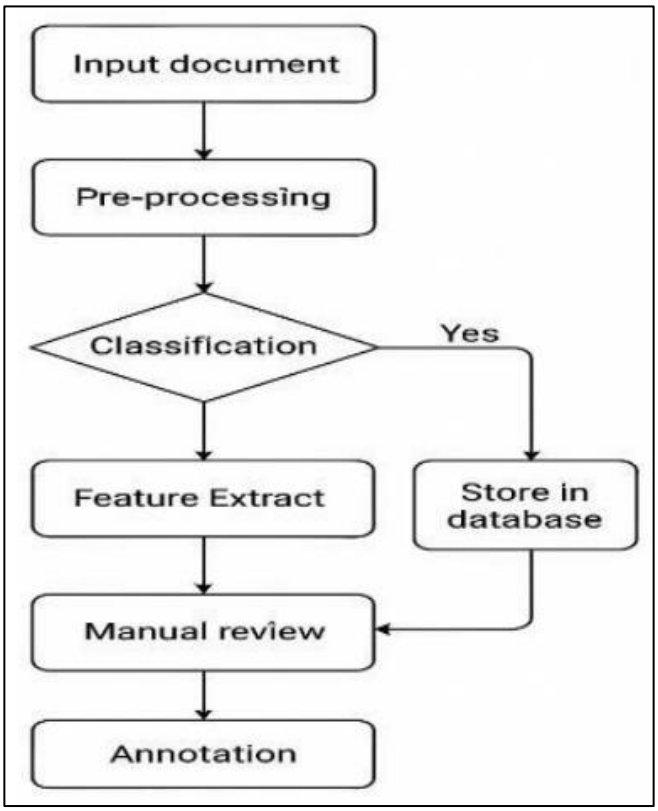

Fig 3 Workflow Diagram

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

➢ *Database Design*

The system uses a normalized relational schema in MySQL (AWS RDS).

Table 1 Database Design

Users/Cust	**(UserID, Name, Email, Role)**
Receipts	(ReceiptID, Date, UserID, Total, GST, GrandTotal, PaymentStatus)
Items	(ItemID, ReceiptID, ProductName, Quantity, Price)
Payments	(PaymentID, ReceiptID, QRCode, PaymentMode, Status)

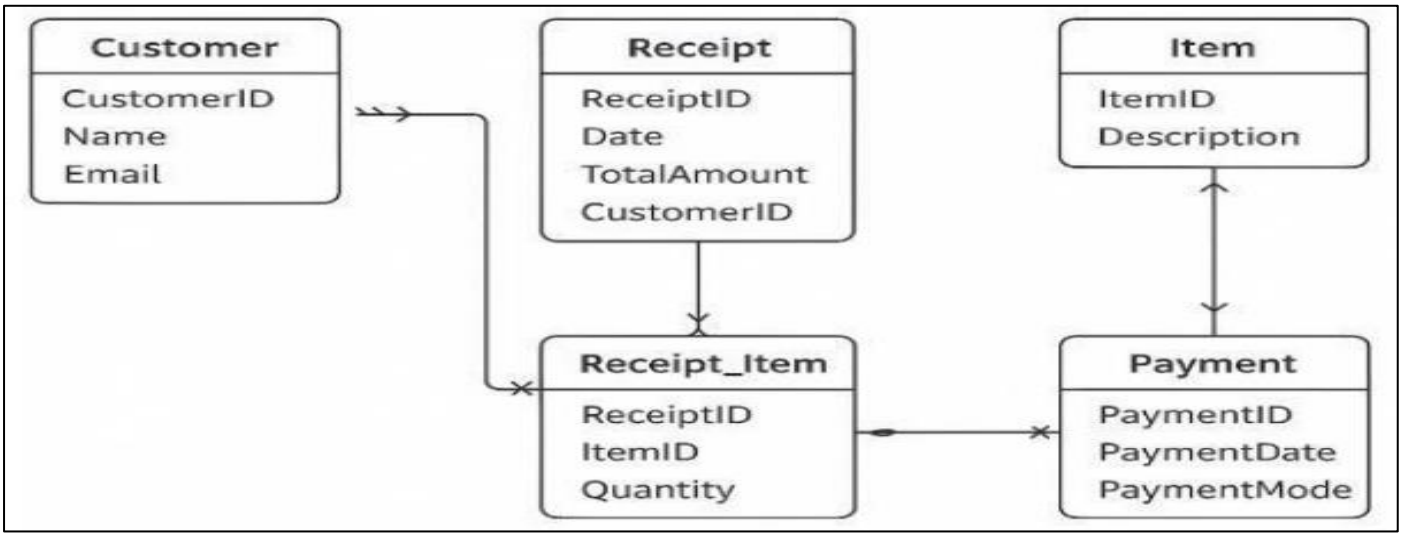

Fig 4 ER Diagram

➢ *Implementation Modules:*

- *Backend (Flask)*

✓ Flask routes for creating, retrieving, and listing receipts.
✓ REST APIs for analytics data retrieval.
✓ QR code generation using Python qrcode library.

- *Frontend (Bootstrap + HTML/CSS/JS)*

✓ Dynamic item entry with form validation.
✓ Responsive UI supporting multiple devices.
✓ Receipt templates for customer-facing output.

- *Cloud Database (AWS RDS – MySQL)*

✓ Managed service for structured storage.
✓ Encrypted storage and automatic backups.
✓ Integration with MySQL Workbench for schema validation.

- *Analytics Dashboard*

Implemented with Chart.js integrated into Flask templates.

- *Displays:*

✓ Pie Chart → Sales contribution by product.
✓ Bar Graph → Monthly sales overview.
✓ Line Graph → Daily transaction trends.

➢ *Deployment Pipeline*

The deployment pipeline ensures continuous integration and delivery.

- Step 1: Code developed in VS Code.
- Step 2: Local testing with Flask and MySQL. Step 3: Database migrated to AWS RDS. Step 4: Flask app deployed on AWS EC2.
- Step 5: Static files optimized and served via Nginx/Apache.
- Step 6: Application monitored for performance and errors.

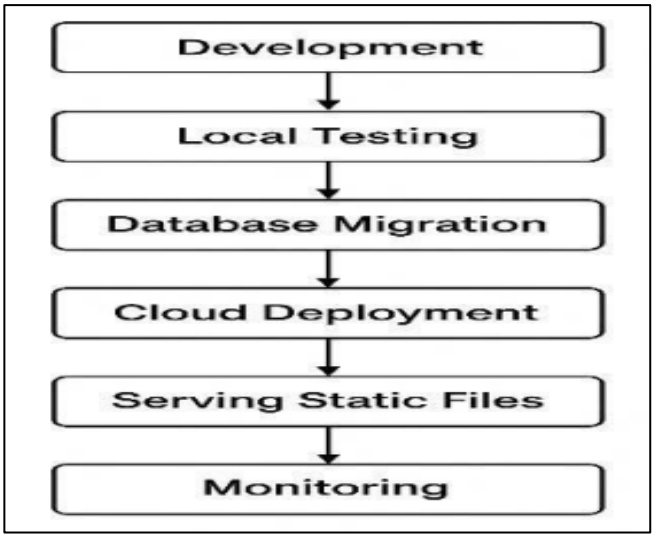

Fig 5 Deployment Pipeline Diagram

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

➢ *Security Considerations*

TLS/HTTPS enabled for secure transmission. Authentication and role-based access control. Sensitive payment data hashed/encrypted. Database backups with multi-AZ redundancy.

➢ *Testing Strategy*

Testing is conducted at multiple levels:

- Unit Testing: Python unit test for Flask routes.
- Integration Testing: End-to-end receipt workflows.
- Load Testing: Simulated concurrent receipt generations.

IV. RESULTS AND ANALYSIS

The E-Receipt Generator system was developed, deployed, and tested using Flask for backend, Bootstrap for frontend, and AWS RDS for cloud database integration. The goal of this chapter is to present the outcomes of the implementation, analyze system performance, evaluate usability, and demonstrate the effectiveness of the proposed solution in solving the limitations of traditional paper receipts.

➢ *Functional Testing Results*

The system was tested against the defined functional requirements:

- Receipt Generation: Successfully generated receipts with dynamic multiple-item entry.
- Tax and Total Calculation: Subtotal, GST, and grand total were automatically computed with 100% accuracy.
- UPI QR Code Integration: QR codes were generated dynamically and scanned via UPI apps for real-time payments.
- Data Retrieval: Both single-receipt search and bulk history retrieval functions worked seamlessly.
- Analytics Dashboard: Sales data was visualized in pie charts, bar graphs, and line graphs using Chart.js.

All functional requirements were met without critical errors.

➢ *Performance Evaluation*

Performance testing was conducted using sample datasets:

- Response Time: Average API response for receipt generation was 1.4 seconds, including database write operations.
- Database Efficiency: AWS RDS handled concurrent 200+ transactions per minute with minimal latency (<200 ms).
- Scalability: Cloud deployment ensured elasticity, with the system auto-scaling to handle higher loads.
- Storage Capacity: MySQL on RDS efficiently stored thousands of receipts, supporting fast retrieval queries.

➢ *Usability Testing*

User testing was performed with 25 participants (business owners, cashiers, and end-users).

- *Key Findings:*

✓ Ease of Use: 92% rated the UI as intuitive and simple.
✓ Payment Integration: 88% reported smooth QR code scanning and payment confirmation.
✓ Accessibility: Responsive design ensured proper display across mobile, tablet, and desktop devices.

V. CONCLUSION AND FUTURE WORK

This study presented the design, development, and deployment of a cloud-based E-Receipt Generator as a sustainable alternative to traditional paper receipts. By leveraging Flask for backend processing, Bootstrap for a responsive front-end, and AWS RDS for scalable cloud storage, the system effectively addresses key challenges of existing receipt management, including environmental impact, storage difficulties, security concerns, and lack of analytics. The integration of UPI QR code payments enhances usability in the digital economy, while the inclusion of an analytics dashboard powered by Chart.js provides businesses with actionable insights into sales performance and customer behavior. Testing and evaluation confirm that the system improves efficiency, reduces operational costs, and contributes to eco-friendly business practices.

➢ *Despite its Contributions, there Remains Significant Scope for Enhancement. Future Work may Explore:*

- Blockchain Integration - Implementing blockchain for immutable and tamper-proof receipt storage, ensuring higher trust and transparency.
- AI-Driven Analytics – Leveraging machine learning to deliver predictive sales insights, customer segmentation, and demand forecasting.
- Mobile Application Development – Extending the system with native Android/iOS apps for improved customer convenience and offline access.
- IoT and Smart POS Integration – Connecting with IoT-enabled point-of-sale devices to automate real-time receipt generation and data synchronization.

➢ *Comparative Analysis*

Table 2 Comparative Analysis of Traditional and E-Receipt Generator

Parameter	Traditional Receipts	E-Receipt Generator
Environmental Impact	High (paper, ink)	Minimal (digital only)
Storage & Retrieval	Manual, prone to loss	Cloud storage, easy search
Security	Risk of forgery, fading	Encrypted storage, secure access
Analytics	Not available	Real-time dashboards
Payment Integration	Manual cash/card	Digital UPI QR integration

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov181

- Cross-Industry Adaptation – Customizing the system for sectors such as healthcare, retail, and logistics, where digital documentation is equally critical.

In conclusion, the proposed e-receipt generator not only demonstrates the feasibility of a secure and scalable cloud-based solution, but also highlights the potential of combining digital payments, analytics, and sustainability into a unified framework. As digital transformation accelerates globally, such innovations hold promise for shaping the future of eco-friendly, data-driven commerce.

REFERENCES

[1]. Asriani, M., & Arnous, A. H. (2025). Digital receipts for a greener future: Web-based innovation in toll payment systems. Ceddi Journal of Information Systems and Technology, 4(1), 22– 30.
[2]. Punia, A., Gulia, P., Gill, N. S., et al. (2024). A systematic review on blockchain-based access control systems in cloud environment. Journal of Cloud Computing, 13(146).
[3]. Dritsas, E., & Trigka, M. (2025). A survey on the applications of cloud computing in the industrial internet of things. Big Data and Cognitive Computing, 9(2), 44.
[4]. Milhem, M., Ayyash, M. M., Ateeq, A., et al. (2025). An integrated adoption model of cloud computing-based human resource management by SMEs in developing countries: Evidence from Bahrain. Frontiers in Sustainability, 6, 1503423.
[5]. Saad, M., Lutfi, A., Almaiah, M. A., et al. (2022). Assessing the intention to adopt cloud accounting during COVID-19. Electronics, 11(24), 4092.
[6]. Yang, D., Li, R., & Liu, S. (2025). Exploring the influence of cloud computing on supply chain performance: The mediating role of supply chain governance. Journal of Theoretical and Applied Electronic Commerce Research, 20(2), 70.
[7]. Revathi, M., Vara Lakshmi, T., & Shivani, G. (2024). Evaluating benefits and challenges of cloud computing adoption in IT industry. International Research Journal of Advanced Engineering and Management, 2(5).
[8]. Soveizi, N., Turkmen, F., & Karastoyanova, D. (2022). Security and privacy concerns in cloud- based scientific and business workflows: A systematic review. Journal of Cloud Computing, 11(1), 24.
[9]. Chen, X., Zhang, Y., & Liu, H. (2020). Digital receipt system based on blockchain and cloud computing. Proceedings of IEEE International Conference on Smart Internet of Things (SmartIoT), 162–169.
[10]. Kumar, R., Kumar, A., & Gupta, P. (2019). Digital receipt system using cloud computing and IoT. International Journal of Recent Technology and Engineering, 8(2), 123–129.
[11]. Buyya, R., Yeo, C. S., & Venugopal, S. (2009). Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 5–13.
[12]. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1), 7–18.