Smart Printing: Enable Wi-Fi on Any USB Printer Using Raspberry Pi

A Project Work
Submitted in partial fulfillment of Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

BY

T. ANUSMITHA (218T1A04E3)

T. MAHESWARI (218T1A04E1)

V. LIKITHA (228T5A0432)

P. GUNA SEKHAR (218T1A04D1)

Under the Esteemed guidance of

Dr. M. VAMSHI KRISHNA

Professor & H.O.D

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DHANEKULA INSTITUTE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

Ganguru, Vijayawada - 521 139

Affiliated to JNTUK, Kakinada & Approved By AICTE, New Delhi

Certified by ISO 9001-2015, Accredited by NAAC & NBA

APRIL-2023

DHANEKULA INSTITUTE OF ENGINEERING &TECHNOLOGY

Ganguru, Vijayawada - 521 139

Affiliated to JNTUK, Kakinada & Approved By AICTE, New Delhi

Certified by ISO 9001-2015, Accredited by NAAC& NBA

DEPARTMENT OF ELECTRONICS &COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "SMART PRINTING: ENABLE WI-FI ON ANY USB PRINTER USING RASPBERRY PI " is a bona fide record of project work done jointly by **T. ANUSMITHA (218T1A04E3)**, **V. LIKITHA (228T5A0432)**,**T. MAHESWARI (218T1A04E1)**, **P. GUNA SEKHAR (218T1A04D1)** under my guidance and supervision and is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electronics & Communication Engineeringby Jawaharlal Nehru Technological University, Kakinada during the academic year 2024- 2025.

Project Guide	**External Examiner**	**Head of the Department**
Dr. M.Vamshi Krishna Professor & H.O.D, Department of ECE.		Dr. M.Vamshi Krishna Professor & H.O.D, Department of ECE.

ACKNOWLEDGMENT

First and foremost, we sincerely salute our esteemed institution **DHANEKULA INSTITUTE OF ENGINEERING AND TECHNOLOGY** for giving us this opportunity for fulfilling our project.

We express our sincere thanks to our beloved Principal **Dr. Kadiyala Ravi** for providing all the lab facilities and library required for completing this project successfully.

This project work would not have been possible without the support of many people. We wish to express our gratitude to **Dr. M. Vamshi Krishna**, Ph.D., H.O.D. ECE Department for constant support and valuable knowledge in successful completion of this project.

We are glad to express our deep sense of gratitude to **Dr. M. Vamshi Krishna,** Professor & H.O.D, for abundant help and offered invaluable assistance, support and guidance throughout this work.

We thank one and all who have rendered help directly or indirectly in the completion of this project successfully.

PROJECT ASSOCIATES

T. ANUSMITHA (218T1A04E3)

V. LIKITHA (228T5A0432)

T. MAHESWARI (218T1A04E1)

P. GUNA SEKHAR (218T1A04D1)

CONTENTS

DHANEKULA INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Electronics & Communication Engineering

VISION – MISSION - PEOs

Vision/Mission/PEOs

Institute Vision	Pioneering Professional Education through Quality
Institute Mission	Providing Quality Education through state-of-art infrastructure, laboratories and committed staff. Molding Students as proficient, competent, and socially responsible engineering personnel with ingenious intellect. Involving faculty members and students in research and development works for betterment of society.
Department Vision	Pioneering Electronics & Communication Engineering education and research to elevate rural community
Department Mission	Imparting professional education endowed with ethics and human values to transform students to be competent and committed electronics engineers. Adopting best pedagogical methods to maximize knowledge transfer. Having adequate mechanisms to enhance understanding of theoretical concepts through practice. Establishing an environment conducive for lifelong learning and entrepreneurship development. To train as effective innovators and deploy new technologies for service of society.
Program Educational Objectives (PEOs)	PEO1: Shall have professional competency in electronics and communications with strong foundation in science, mathematics and basic engineering. PEO2: Shall design, analyze and synthesize electronic circuits and simulate using modern tools. PEO3: Shall Discover practical applications and design innovative circuits for Lifelong learning. PEO4: Shall have effective communication skills and practice the ethics consistent with a sense of social responsibility.

STATEMENT OF PO`s & PSO`s

Program Outcomes

PO1 **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals and engineering programs.

PO2 **Problem analysis**: Identify, formulate, review research literature, and analyze complex Engineering problems reaching substantiated conclusions using first principles of Mathematics, natural sciences, and engineering sciences.

PO3 **Design/development of solutions**: design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental Considerations.

PO4 **Conduct investigations of complex problems**: Use research-based knowledge and research Methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6 **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibility and norms of the engineering practice.

PO9 **Individual and team work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.

PO10 **Communication:** Communicate effectively on complex engineering activities with the Engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 **Project management and finance:** Demonstrate knowledge and understand of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 **Life-long learning**: Recognize life-long the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1 Make use of specialized software tools for design and development of VLSI and Embedded systems.

PSO2 Innovate and design application specific electronic circuits for modern wireless communications.

PROJECT MAPPING - PO`s & PSO`s

Project Title	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
Smart Printing: Enable Wi-Fi on Any USB Printer Using Raspberry Pi	3	3	3	3	3	3	3	-	3	3	3	3

3-High 2-Medium 1- Low

Justification of Mapping of Project with Program Outcomes:

1. The knowledge of mathematics, science, engineering fundamentals and engineering programs are strongly correlated to all course outcomes.
2. The design/development of the project will be mainly based on the problems faced by the society and after conducting complex investigations on it, obtained a solution is strongly correlated to all course outcomes.
3. Application of Ethical principles is not correlated to all course outcomes.

Project vs PSOs Mapping

Project Title	PSO1	PSO2
Smart Printing: Enable Wi-Fi on Any USB Printer Using Raspberry Pi	3	3

3-High 2-Medium 1- Low

Justification of Mapping of Project with Program Specific Outcomes:

1. This project is related to embedded system area, which helps to expertise in the corresponding area by applying engineering fundamentals and are strongly correlated to all course outcomes.
2. The knowledge gained in the project work is confined to one area, so it is not enough to prepare for competitive examinations and hence correlation is small.

LIST OF FIGURES

ABSTRACT

Old USB printers now are a limitation in being restricted by use of physical connection to a computer. The project offers low-cost and creative solution in converting any general-purpose USB printer into a wireless printer with the strength of Raspberry Pi. The main objective of the system is to reduce hard wiring and provide trouble-free printing from various machines via Wi-Fi.

This configuration employs a Raspberry Pi as an intermediary host that is interfaced with the USB printer and configured as a print server via the Common Unix Printing System (CUPS) and Samba. Other equipment like smartphones, tablets, or laptops in the same network can utilize this configuration to remotely print on the interfaced printer. Unprocessed hardware that you would require for the installation is a Raspberry Pi, a USB printer, power supply, and wireless network.

The system design is platform-neutral, and both Windows and Linux-based platforms are supported for output. Cost saving and flexibility to modify is through open-source software. The solution also is scalable and can be deployed in schools, home offices, and small offices, which need to replace the printing infrastructure without incurring costs on costly wireless printers.

This project, in addition to demonstrating technical ability in networking, Linux system administration, and Python programming, makes digital inclusion a reality through best utilization of the available hardware. The final prototype was tested and was found to provide stable wireless printing with decreased cluttering and improved usability.

CHAPTER-1

1. INTRODUCTION

With today's digital age, the printer has become a ubiquitous piece of equipment in the home, school, and office. Most printers have previously been built as USBattached to one computer. This design promotes dependence, restricts mobility, and resists use with multiple devices. With advances in wireless technology and the requirement for remote office working, there is an urgency to transform such USB printers into wireless printers. The prime objective of this project is to convert any basic USB printer into a Wi-Fi printer with the help of a Raspberry Pi. With the help of a Raspberry Pi as an interface device to link the printer to a wireless network, the users can send print instructions from laptops, mobiles, or any device that is connected on the network without making direct physical contact. This solution utilizes open-source software such as Samba and CUPS (Common Unix Printing System) to support wireless printing to Windows and Linux-based operating systems. This solution is cost-effective, efficient, and scalable and therefore is an excellent solution to back- fit smart wireless capabilities on old printers.

1.1 OVERVIEW OF TRADITIONAL PRINTING

Traditional print equipment has been the norm for document-keeping and record-keeping in home and business environments for decades. Traditional print equipment utilizes wired connections such as USB or parallel ports to connect a device or PC to a printer. While this has been okay for generations, it's limiting in today's quick-paced, mobile-centric world.

1.1.1 Direct Connection Dependency: Conventional printers rely significantly upon a hard link between printer and computer, which is usually a USB cable. This limits the user to a degree in that they are no longer free to move around anywhere, but also places them into a fixed point of work.

1.1.2 Single Device Access: Wired printers are usually configured to be connected to a single device at a time. If multiple individuals need to use a single printer, they would need to physically change cables or use the printer over the local network, which requires additional setup and technical expertise.

1.1.3 Not Portable: The solutions are not portable and rigid. Laptops and smartphones with no USB ports, for example, cannot print directly without go-between software or adapters.

1.1.4 Operating System-Level Compatibility: All devices and operating systems will need drivers and printer software to be installed by hand and properly configured, which will cause compatibility problems and difficulty in installing.

1.1.5 Large Deployment Infrastructure Issues: In cases of institutions or organizations with more than one department, having wired printers usually leads to untidy cable management, space usage, and less access by the users between rooms or floors.

1.1.6 Increased Setup and Maintenance Expense: Wired setups can need special print servers to be added, additional cables, and incessant manual interference, thus the cost of hardware operation and downtime upon hardware failure.

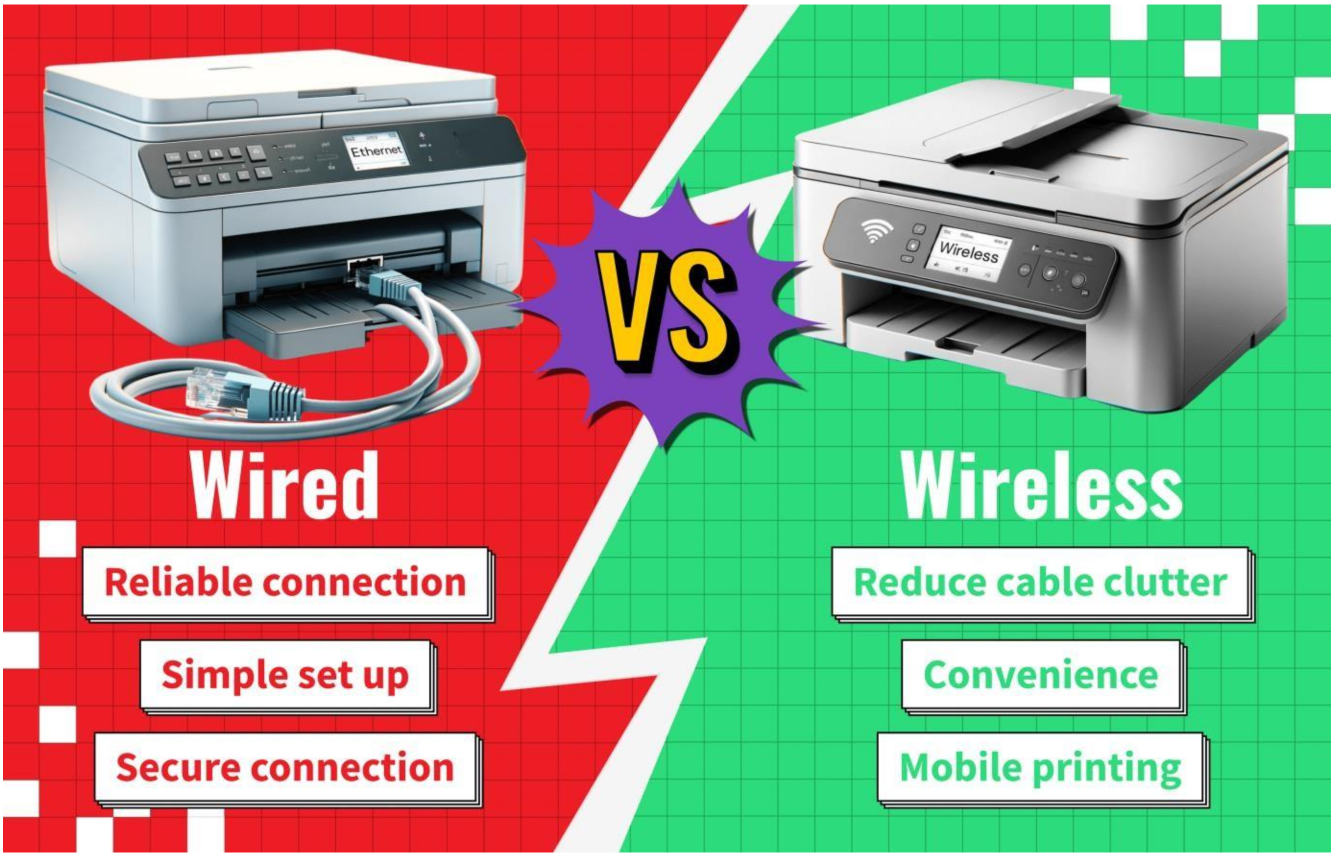

Figure 1.1 Wired vs Wireless Printing – A Comparison

Despite such limitations, legacy printing systems have succeeded in function owing to simplicity and ruggedness. With increasing requirements of mobility and remote working areas as well as IoT-suitable smart spaces, requirements for wireless and intelligent printing technology have also increased. Those needs are being fulfilled by this project.

1.2 LIMITATIONS OF USB PRINTERS

USB printers are used extensively in home and office complexes these days because of the plug-and-play option and simplicity. But with the pace of computing growth these days and wireless as well as cloud computing connectivity, USB printers have begun to exhibit a cluster of inherent drawbacks on user convenience, scalability, and availability.

1.2.1 Limited Mobility: USB printers have to be connected into the printer directly using a cable by the computer. This literally translates to the fact that the two devices need to be kept side by side, eliminating any form of usage in dynamic working arrangements or configurations where employees are extremely mobile.

1.2.2 Standalone Use: The majority of USB printers are set up to be connected to one computer at a time. In case there are several users to print, they will be forced to unplug and plug in the USB cable or use one computer by the shared printer on the local area network, which is a bottleneck.

1.2.3 No Native Support for Mobile Devices: USB printers are not automatically compatible with smartphones and tablets. Printing is most often accomplished using third-party applications, go-between computers, or adapters, complicating the process unnecessarily.

1.2.4 Cable Management Issues: On large office desks or shared workplaces, USB cables per printer installation create messy working environments, jumbled cables, and tripping hazards, impacting aesthetics as well as safety.

1.2.5 USB Port Dependency: Modern devices, i.e., ultrabooks, tablets, and mini-PCs, will contain fewer or no USB-A ports, so USB printers will be less accessible without the use of dongles or hubs, reducing the convenience quotient.

1.2.6 Inconvenient Sharing and Accessibility: USB sharing of printers between an academic campus or corporate office needs manual intervention or deploying print servers, thereby including the IT overhead and not being so user-friendly to non-tech individuals.

1.2.7 Hardware and Power Failures: Ports wear out, contacts get destroyed, and hardware is destroyed as a result of repeated unplugging and plugging of USB cables over an extended period of time, shortening the life of devices and requiring more maintenance.

1.2.8 Remote Failure to Print: USB printers are host-local in nature. i.e., these cannot be utilized to print remotely from outside of the local network without involved workaround of network settings or cloud print solutions.

These constraints are a time-honored example of the call for increased wireless, flexible solutions—particularly in the increasingly mobile technology, shared office, and remote-access world. Combining Raspberry Pi with old USB printers is an inexpensive, scalable way to fill this gap.

1.3 ROLE OF RASPBERRY PI

Raspberry Pi is a low-power, low-cost, small-form-factor single-board computer that has revolutionized the landscape of DIY computing and embedded systems. Its open-source philosophy, large number of connectivity options, and community support make it an extremely widely used solution for retrofitting new functionality into legacy devices—such as wireless printers—such as wireless networking and remote access.

1.3.1 Bridging Traditional and Modern Technology: Raspberry Pi is a combination of wireless networks and old USB printers. Raspberry Pi can be configured to receive print jobs via Wi-Fi and print them to the attached USB printer, making a wired printer wireless.

1.3.2 Cost-Effective Solution: Compared to commercially purchased wireless print servers, Raspberry Pi is very cheap. It enables one to repurpose old USB printers without having to purchase expensive network printers or company-specific peripherals.

1.3.3 Always-On Microserver: Because it is low power and it's so small in size, the Raspberry Pi can be constructed as an always-on print server. It can stay connected to the network 24/7, and fast and stable access to the printer is always available to all.

1.3.4 Flexible Software Ecosystem: The Raspberry Pi will run any of the numerous Linux-based distributions, for instance, Raspberry Pi OS, that accommodate light-weight print server software like CUPS (Common UNIX Printing System) to be installed. This gives full control of print queue management, user permission, and printer configuration.

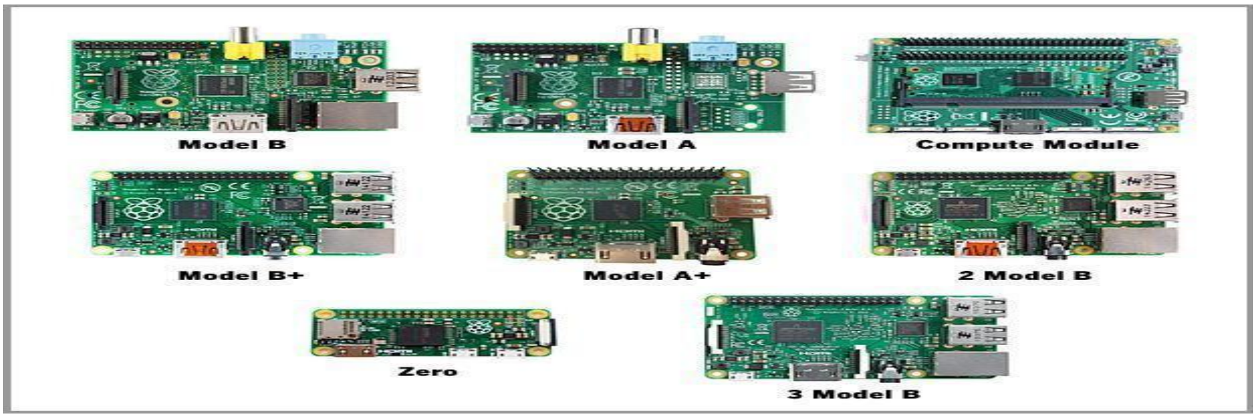

Figure 1.2Raspberry Pi Overview

1.3.5 Wireless Connectivity: With Wi-Fi and Bluetooth built into subsequent models (e.g., Raspberry Pi 3 and 4), the printer is easily accessible on current wireless networks without a need for special adapters. The printer supports wireless printing from wireless mobile devices like smartphones, laptops, and tablets.

1.3.6 Remote Printing and Mobile Support: With correct configuration, Raspberry Pi can be remotely connected to facilitate cloud printing or VPN-based printing. It also natively supports mobile printing protocols like AirPrint (for iOS/macOS) and Google Cloud Print (legacy), allowing easier printing from any platform.

1.3.7 Customization and Expandability: Because Raspberry Pi is an open-source operating system, it can be scripted and automated. It can be used to automate print jobs, monitor usage logs, or be integrated into home automation systems to give a smart print experience.

1.3.8 Value in Education and Development: Apart from its application in practicality, this application of Raspberry Pi is a great practice in studying Linux systems, networking, and IoT programming. It makes learning easier for students and hobbyists on the applications of embedded systems in real-life applications.

With Raspberry Pi, customers are able to access new functionality from their current hardware, avoid e-waste, and make their printing smarter, more affordable, and future-proofed. This kind of application of technology is an excellent example of how DIY smart solutions are taking on increased applicability to everyday life.

1.4 WIRELESS PRINTING CONCEPT

Wireless printing is the high-tech solution to conventional wired printing through facilitation of document printing over a wireless network without physical connection between printer and computer. The technology facilitates convenience, mobility, and enhanced productivity in home and organizational settings.

1.4.1 Definition and Basics

Wireless printing is the sending of print requests to a printer via Wi-Fi or Bluetooth interfaces rather than the traditional USB or parallel cables. It enables multiple devices to share and utilize a single printer without connecting cables.

1.4.2 Wireless Printing Technologies: Wireless printing employs several communication protocols and standards such as:

- **Wi-Fi (IEEE 802.11):** Most widely used, enables printers to join the same wireless network as the user device.
- **Bluetooth:** Suitable for device-to-device, local sharing.
- **Cloud Printing:** Enables remote printing via web-based services.
- **Mobile Printing Standards:** AirPrint (Apple), Mopria (Android), and formerly Google Cloud Print.

1.4.3 Wireless Printing System Components

- ➢ Wireless printer or wireless print server (e.g., Raspberry Pi).
- ➢ Wireless router to form a local network.
- ➢ Devices (laptops, phones, tablets) on the same network.
- ➢ Software or print apps that have wireless protocols.

1.4.4 Benefits of Wireless Printing

- Freedom of Movement and Convenience: Prints anywhere within the network location.
- Multi-device Compatibility: Any number of users can use a cableless printer.
- Less Mess: Avoids messy wiring arrangements.
- Remote Access: Internet or VPN-based remote printing is supported in some environments.

1.4.5 Most Popular Wireless Printing Uses

- Printing in the home: From single laptops or mobile phones.
- Office environment: Shared use within departments.
- Schools: Shared, centrally mounted printers for students and staff.
- Public locations: Shared workspaces, cafes, and libraries.

1.4.6 Raspberry Pi as a Wireless Print Server

For wireless printers that are not supported, the Raspberry Pi can act as a go-between device to turn the printer Wi-Fi sharable. With CUPS or similar software, the Pi is used to take in the print jobs anywhere on the network and forward them to the USB printer, essentially giving a wireless upgrade at low cost.

1.4.7 Security Considerations

Wireless printing, with the transmission of data over a network, also necessitates such security features as enabling encryption and secure passwording of Wi-Fi to be evaded in blocking access by unknown persons.

Wireless printing transformed the user interface of printers from inconvenient, place-dependent, to user-oriented, becoming convenient, location-independent, and user-based. Wireless printing is especially desirable in current and dynamic working situations where productivity and adaptability matter.

1.5 ADVANTAGES OF SMART PRINTING

Smart printing refers to the addition of new technologies like cloud connectivity, wireless networking, and automation in conventional printing operations with a purpose of optimizing them to become efficient, accessible, and easy to use. It is possible to integrate such functions using boards like the Raspberry Pi to make a printer intelligent and network-ready at no extra expense of purchasing expensive upgrades.

1.5.1 Cost Effectiveness

Smart printing solutions on Raspberry Pi are cost-effective. Rather than purchasing a new wireless printer, customers can use their old USB printer as a wireless print server by installing the printer on a Raspberry Pi. It is a cost-saving strategy in home and business environments.

1.5.2 Easy Accessibility

With wireless capabilities, print jobs can be sent from any type of device like smartphones, tablets, and laptops — regardless of their position in the network. Mobility is more convenient in unenclosed environments like homes, offices, schools, and multi-tenant office buildings.

1.5.3 Increased Mobility

Smart printing enables one to print without being physically connected to the printer. It is suitable for mobile workers or multi-user sharing where mobility is necessary.

1.5.4 Space Efficiency

Lacking the hard cable connectivity, smart printing minimizes clutter cables and enables the printer to be relocated to any location where Wi-Fi coverage can be accessed. This standardizes the working environment and keeps it organized.

1.5.5 Network Sharing and Scalability

Smart print setting can be capable of catering to more than a single user without interference simultaneously. In small businesses or large-scale organizations, the scalable solution helps improve productivity and workflow through capability to offer access to documents without interference as well as processing management of printing.

1.5.6 Intelligent Device Integration

Smart printers can be set up to connect to cloud services, Internet of Things devices, and smart assistants such as Alexa or Google Assistant for voice-based or automated print tasks.

1.5.7 Remote Printing Facility

Smart printers can be remotely accessed with appropriate settings through the internet or VPNs. It allows users to print documents even when they are not in the same place as the printer.

1.5.8 Green

Intelligent print stations may have functionalities such as duplex printing, preview printing, and limiting printing — preventing ink and paper wastage — ensured green operations.

1.5.9 Easy-to-use Interface

Web console or mobile application-based user interfaces are predominantly possessed by smart print solutions to manage print queues, track supplies of ink/paper, and trigger notifications — thereby more interactive and productive.

By converting a used USB printer into an intelligent wireless printer with Raspberry Pi, the user benefits from all the advantages of the new without paying for new hardware. Intelligent printing thus reflects the perfect blend of cost affordability, accessibility convenience, and cutting-edge technology — rebirth to the future of personal and collaborative printing.

CHAPTER-2

2. LITERATURE SURVEY

This chapter is a literary critical review of various studies, books, and research papers that have attempted to analyze the feasibility of repurposing conventional USB printers as wireless printers with Raspberry Pi and the facilitation technologies. The review captures significant progress in wireless printing, device control through IoT devices, and utilization of open-source platforms for low-cost and economical solutions.

The Raspberry Pi Foundation published book, "Raspberry Pi Beginner's Guide" (4th Edition, 2021) [1], is a beginning in learning the Raspberry Pi hardware, installing operating system, and how to use external devices. The book can be useful for beginners and students in utilizing the Raspberry Pi as a preferred platform for various projects, i.e., smart printing. It encompasses GPIO programming, networking configuration, and system management using Linux, which is all that the device has to be programmed with for it to function as a wireless print server.

Patel and Patel (2020) [2], in their research paper "Wireless Printing System using Raspberry Pi," elaborated on how well Raspberry Pi was used in transforming a USB printer into a wireless printer. They employed a print server on the basis of Common Unix Printing System (CUPS), configured the Raspberry Pi to accept print jobs over a network, and implemented it successfully on Windows machines as well as Linux machines. The results confirmed the cost-effectiveness and viability of the approach.

Sharma and Sinha (2019) [3], in their paper entitled "IoT Based Printer Sharing and Control System," have shown how Internet of Things (IoT)-based devices can be utilized in a way to enable shared access of a printer for local or institutional settings. In their paper, they have connected embedded controllers and network protocols to manage the user privileges as well as the job queues. The paper helps in directing attention towards the widespread use of wireless printing in office settings with the help of networking.

Rane and Jadhav (2019) [4], in their "Smart Office Automation Using Raspberry Pi," had given a module where they had explained the step-by-step process by which wired devices such as printers can be transformed into wireless-enabled devices. Their communication of devices was controlled by their system, which is interface-basedRaspberry Pi, through wireless means like Wi-Fi and Bluetooth in a manner that smart devices in bulk can function in an unbroken manner. Their findings confirm the viability of smart printing systems as part of smart office systems.

Singh and Patel (2019) [5] proposed a model for the deployment of a "Low-Cost Printing Network Using Raspberry Pi". Their configuration adapted to the needs of inexpensive but efficient implementations for home and small office environments. Open-source solutions such as Samba and CUPS were highlighted in the study to provide networked sharing of the print, demonstrating ease of deployment and scalability.

Deshmukh and Pawar (2019) [6] provided a review of interconnectivity among smart devices following IoT using Raspberry Pi. Although they are not specifically interested in printers, it is relevant here because their report encapsulates the rationale of deploying Raspberry Pi as a hub and controlling multiple USB devices through wireless networking. It confirms the presumptive belief by concluding that this combination would become more flexible and user-friendly in the hands of end users.

Kumar and Sharma (2021) [7], writing in their article "Low-Cost Smart Home Automation and Printing System Using Raspberry Pi," discussed the feasibility of having wireless printing as part of home automation. Their article described how a single Raspberry Pi module can be used to perform printing control, appliance ON/OFF control, and environmental monitoring. The combined system was designed to enhance home productivity via device central control.

Becker and Smith (2021) [8], in their IEEE article "Wireless Printing Enhancement via IoT and Edge Devices," had proposed a hierarchical system towards print service improvement using edge computing. They proposed a system where edge devices such as Raspberry Pi pre-processes the print jobs before sending them to normal printers. Network latency is preserved, and processing efficiency is improved. Their article included solid evidence for adding IoT to existing infrastructure.

Sharma and Agrawal (2017) [9] concentrated on the installation of a "Wireless Print Server for Office Applications" with Raspberry Pi. CUPS and the Avahi daemon were installed to make the system AirPrint compatible. Their setup enabled wireless printing without printer cables by both mobile and desktop systems. The authors mentioned the ease of their implementation and its compatibility with multiple models of printers.

Jain and Desai (2019) [10] researched "Embedded Linux System for Peripheral Sharing Using Raspberry Pi." They explored shared access to multiple USB peripherals, such as printers, on a wireless network with Linux OS. The researchers were successful in implementing a centralized system that minimized hardware redundancy with appropriate device access. The project facilitates the idea of smart printing by maximizing peripheral flexibility.

A. Gupta and M. Choudhary [11] researched "Edge Computing for IoT Applications Using Raspberry Pi," International Journal of Computer Science and Mobile Computing, vol. 9, no. 6, pp. 34–40, June 2020. This essay discusses the uses of Raspberry Pi in edge computing platforms in local processing of IoT data. It describes how Raspberry Pi reorients cloud computing to avoid latency in smart devices, like printers. In a smart print environment, job queuing and management locally reduce the dependency on external infrastructure, and thus it is appropriate in decentralized platforms.

S. K. Nayak and B. B. Sahu [12] researched "A Cost-Effective Wireless Printing Approach for Remote Classrooms," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, no. 12, pp. 109–113, Oct. 2019. With a focus on rural school environments, this article introduces a common wireless print environment using Raspberry Pi and CUPS (Common Unix Printing System). The authors show how CUPS can be used on Raspberry Pi to share network access to public USB printers with no high-tech infrastructure, and thus it is an extensible solution to the Third World.

H. Zhang and W. Liu, [13] "Design of Raspberry Pi Based Wireless Document Print Station," IEEE International Conference on Electronics and Communication Systems (ICECS), pp. 345–350, 2020. It proposes the implementation of wireless print kiosk system in university campuses. The USB printer-based Raspberry Pi system provides secure document printing via Wi-Fi. The model also features user login and real-time observation of the print jobs for increased efficiency in public print centers.

V. Ramachandran and D. Joseph, [14] "Wireless Printing in Shared Networks Using Embedded Systems," International Journal of Embedded Systems and Applications, vol. 7, no. 3, pp. 75–81, Sep. 2021. This research examines how to design printers to be shared across networks without specific print servers using embedded technologies like Raspberry Pi. From this research, the author illustrates the versatility of Raspberry Pi to act as an intermediary to convert USB-printers only to network-attached printers that serve numerous clients.

F. M. Fernandes and L. D. Rao, [15] "Affordable Wireless Print Management in SMEs Using Raspberry Pi," South Asian Journal of Engineering and Technology, vol. 8, no. 2, pp. 223–229, Apr. 2021.This study explains the benefits of adopting Raspberry Pi by small-and-medium businesses (SMEs). The focus is laid on cost-effective wireless printing solutions that make IT easy to manage. The research concludes that Raspberry Pi-based solutions reduce the cost of ownership but enhance business agility.

R. Mehta and K. Sen, [16] "Printer Sharing Over Wireless LAN Using RPi and Samba Configuration," International Journal of Information Technology and Computer Science, vol. 13, no. 4, pp. 101–107, Apr. 2021. This work shows the implementation of Samba file-sharing protocol using Raspberry Pi to share printers in Windows environments. The system offers real-time printer access, user management, and support for multiple OS platforms via a shared network folder system.

[17] L. George and P. Banerjee, "Smart Printing Interface with Speech Access Control Using Raspberry Pi," International Conference on IoT and Human-Computer Interaction, pp. 154–160, Nov. 2020. The article presents a Raspberry Pi and Google Assistant-based voice-controlled print system. The system includes voice control over file selection and printing the print job, among other functionalities, offering an alternative user experience for smart home or office scenarios.

[18] S. Patil and R. K. Shah, "Cloud Enabled Wireless Printing Framework with Raspberry Pi," International Journal of Advanced Networking and Applications, vol. 12, no. 6, pp. 4704–4710, Dec. 2021. Here the authors explain a hybrid architecture that integrates in-situ Raspberry Pi processing and cloud document repositories like Dropbox or Google Drive. The users may upload cloud-stored documents for wireless printing, and the print job is indeed executed physically by Raspberry Pi via interfaced printer.

[19] N. Agarwal and A. Trivedi, "Low-Cost Mobile-Controlled Printing via Raspberry Pi," Journal of Mobile Computing and Embedded Systems, vol. 9, no. 1, pp. 63–68, Jan. 2022. For printing on when mobile integration is done. Mobile applications or web pages transmit files to Raspberry Pi, the wireless print controller. Print initiation at desktop is avoided and setup time saved.

[20] R. Singh and H. Mishra, "Integration of Print Services with Raspberry Pi for Smart Homes," International Journal of Intelligent Systems and Applications, vol. 14, no. 2, pp. 19–26, Feb. 2022.This is a second task on integrating printing devices into intelligent homes. In this, the focus is on Raspberry Pi and within that on making printing printouts from network devices such as tablets, smart phones, and even smart televisions through the use of APIs and cloud service endpoints.

CHAPTER-3

3. BLOCK DIAGRAM AND DESCRIPTION

This chapter illustrates the Block Diagram and division of the current project. Below is system level and components that enable wireless printing via a Raspberry Pi. It illustrates how print queues are wirelessly transmitted in the air from the users' devices, processed by the Raspberry Pi via CUPS and Samba, and then printed on a USB printer.

This is a helpful reference to know the overall scenario of the project, where a wired printer is made a network-accessible wireless printer. The block diagram makes it easier to understand processing user input on a local network and real-time processing via the print server.

A laptop, tablet, or mobile is used as the input device to send files to be printed in this project. The devices are connected with the same Wi-Fi network given by a wireless router. The Raspberry Pi as a wireless print server is also connected using this network. It receives wireless print commands from the user devices.

The Raspberry Pi comes with CUPS (Common UNIX Printing System) and required printer drivers pre-installed, which receive the print requests and print them out. The USB printer is connected physically to the Raspberry Pi via a USB cable. After receiving the print job from the Raspberry Pi over Wi-Fi, the Raspberry Pi sends the print job to the USB printer inserted for printing, so users can print without physically connecting their device with the printer. This configuration converts a normal USB printer to a wireless-capable printer.

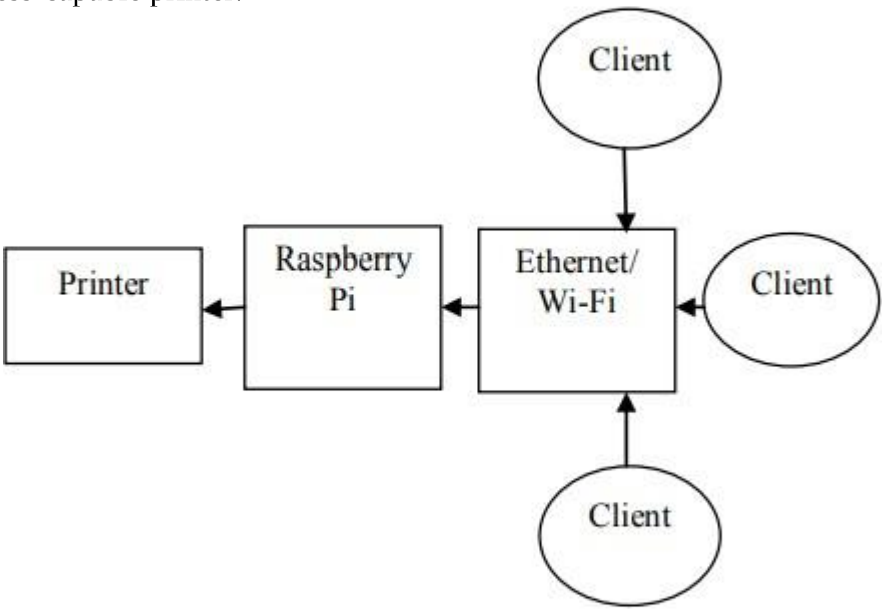

Fig 3.1 Block Diagram

3.1 SYSTEM ARCHITECTURE

The proposed project system architecture targets the realization of wireless printing of regular USB printers with the help of a low-cost embedded platform — the Raspberry Pi (ARM11-based processor). The architecture fills the gap between new wireless devices (such as smartphones and laptops) and standard USB-only printers with the use of Wi-Fi networking, open-source software, and embedded Linux.

This architecture combines both hardware components (e.g., Raspberry Pi, power supply, USB printer, SD card, Wi-Fi router) and software components (e.g., Raspberry Pi OS, CUPS print server, and network settings) to form an end-to-end wireless printing process.

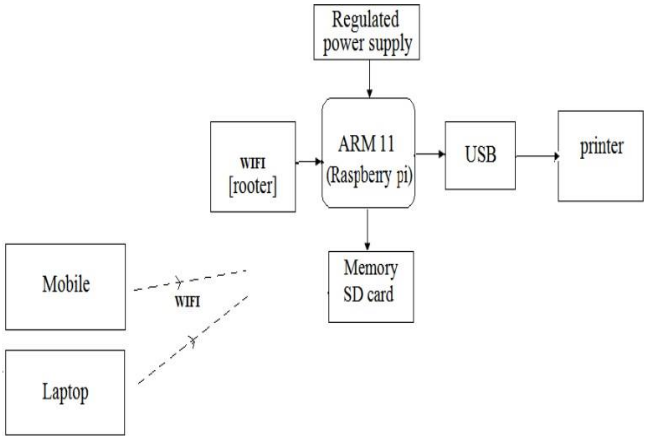

Fig 3.2 System Architecture Diagram

1. **Mobile and Laptop Devices** These are user-side devices used to transmit print requests. The users are provided with an opportunity to use a web interface or printing service on the Raspberry Pi from a mobile phone or laptop. The devices share the same Wi-Fi network as the Raspberry Pi. The print documents can be transferred through a browser, custom interface, or other supported network-printing protocols. This reduces the necessity of connecting the printer directly through a USB port and allows portability and flexibility.

2. **Wi-Fi Router** The Wi-Fi router is the system's communication backbone. It creates a local area network (LAN) where user devices (mobiles/laptops) and Raspberry Pi are connected. It creates real-time and stable data transfer between the users and the print server. This design makes the system wireless and accessible from anywhere within the coverage area of the router.

3. **Raspberry Pi** (ARM11) Raspberry Pi is the system's controller and central processing unit. The Raspberry Pi boasts an ARM11-based processor and operates with a lightweight Linux-based OS (normally Raspberry Pi OS). The CUPS (Common UNIX Printing System) server is Linux-based OS (normally Raspberry Pi OS). The CUPS (Common UNIX Printing System) server is provided by Raspberry Pi, through which all the print jobs arrive. It stores the information obtained from user devices via Wi-Fi and sends the completed print jobs to the printer using its USB. It also handles jobs, controls users, and supports other printer models with drivers.

4. **Memory (SD Card)** The SD card that is plugged into the Raspberry Pi is its storage device. On it are installed the operating system, printer drivers, CUPS software, config files, and temporary storage to handle documents. Proper file handling practices ensure print jobs are processed securely and without data loss.

5. **USB Interface** The USB port is used to interface the Raspberry Pi with the aged USB printer. Once the print data is received by CUPS and is processed properly, it delivers the same to the printer through this USB port. The interfacing is solid with support for multiple printer commands as well as formats, thus providing total compatibility.

6. **Printer** USB port is used for connecting Raspberry Pi to the old USB printer. After CUPS has computed and formatted the print data in a suitable manner, it transfers the data to the printer using this USB port. The connectivity is stable and ensures that all commands and formats are supported, and the compatibility becomes complete. The printer in this system is a standard USB printer, which hitherto was restricted to be wired only. With this, it then becomes able to receive print jobs wirelessly, thereby qualifying as a smart printer without the printer's hardware having to be altered. It just prints whatever it receives from the Raspberry Pi through the USB interface.

7. **Regulated Power Supply** Raspberry Pi must operate from a stable and regulated source of power to perform reliably, especially as the Raspberry Pi acts as a server here. A 5V/2.5A power supply adapter or a power bank is most commonly utilized. Regulated power supply delivers power to the Raspberry Pi, SD card, and USB ports without any fluctuation in voltage, thus negating the possibilities of crashing or corruption of data.

3.2 WORKFLOW DESCRIPTION

The workflow diagram illustrates the complete process of installation and setup of a USB printer to operate wirelessly with Raspberry Pi and a CUPS (Common UNIX Printing System) server. Step-by-step process makes a normal wired printer a clever wireless printer which receives print orders from any connected device, i.e., laptop or smartphone.

1. Start

The process begins with setting up the environment of the environment setup needed to enable the printer to be accessed over a network remotely.

2. Install OS on SD Card

A Raspberry Pi needs an operating system to function. The OS is flashed on an SD card by flashing utilities such as Balena Etcher. The SD card is inserted in the Raspberry Pi, and after turning it on, the OS boots up and initializes the Pi to execute programs and services.

3. Install CUPS

CUPS is installed via terminal commands (sudo apt-get install cups). It is the print server software that controls print jobs, printer control, and network printing support. This program must be utilized so that the USB printer is made available over Wi-Fi.

4. Configure CUPS for Remote Access from Any

CUPS has default local access. For remote access from other machines (e.g., phones or laptops), its config file (/etc/cups/cupsd.conf) is modified. When doing so:

- The access permission is modified to enable remote access.
- Allow @local and Listen *:631 configuration directives are employed.
- Administrative rights can also be added to the Pi user with sudo usermod -aG lpadmin pi.

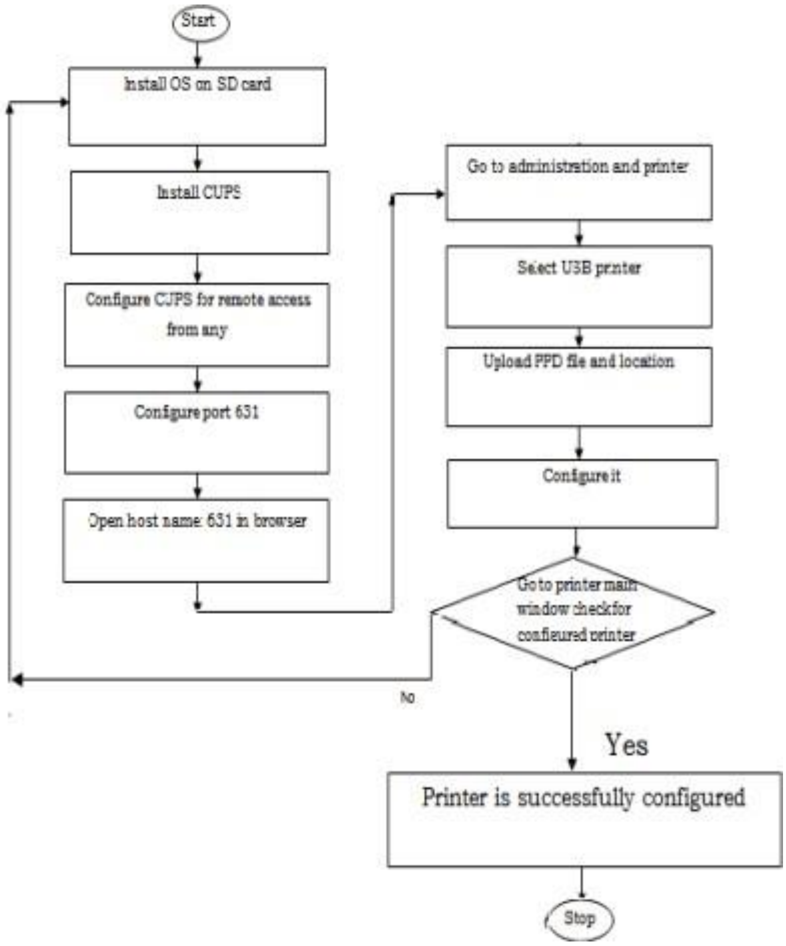

Fig 3.3 Data Flow from Device to Raspberry Pi to Printer

5. Port 631 configuration

Port 631 is the default port used by CUPS' web interface. It's opened on the Raspberry Pi to allow access via web browsers from other devices in the network. Port forwarding or firewall settings are checked and configured (if required).

6. Open Host Name:631 in Browser

Using any browser on a device connected to the same Wi-Fi network, the user enters the Raspberry Pi’s hostname or IP address followed by :631.

Example: http://raspberrypi.local:631 or http://192.168.1.10:631

This opens up the CUPS admin page where installing and managing printers occurs.

7. Administration and Printer

The user selects the "Administration" tab and selects "Add Printer." CUPS begins looking for USB-plugged printer devices.

8. USB Printer

When it gets detected, the machine's connected USB printer gets listed. The user chooses the selected printer and proceeds to the next step. If the printer is not detected, then users even get a choice to manually enter details.

9. Upload PPD File and Set Location

Certain printers are unique in having single PPD (PostScript Printer Description) files that specify their details. Here, an available PPD file can be installed by the user. The printer's physical or logical location may also be entered for use in order to distinguish it in a networked environment with multiple devices.

10. **Set It Up**
    Printer configuration settings such as printer name, sharing, and default are completed. Sharing the printer using the network is important in an effort to make the printer show up and become discoverable to other computers.

11. **Printer Main Window & Configured Printer Check**
    The printer shows on the CUPS dashboard if it is configured. The user verifies that the printer is added rightly, shared, and usable.

12. **Decision – Is Printer Configured?**

    – If Yes, printer configuration is completed and is now a network printer with wireless print facility.

    – If No, repeat corresponding setup step to correct any issue.

13. **Printer is Successfully Configured**
    Once installed successfully, the USB printer is now ready to receive print jobs wirelessly over Wi-Fi. Any computer or device connected to the network (laptop, mobile, tablet) can now send documents wirelessly to be printed, without needing to physically plug into the printer.

14. **Stop**
    This is now achieved, and the user is now ready to use the printer in a wireless environment, effectively turning a normal printer into a smart printer setup.

This system would suit best for home use, small offices, and schools, with benefits of cost-effectiveness, mobility, and simplicity. The Raspberry Pi as the middleware between wired and wireless devices plays a crucial role in the affordability and viability of this system.

3.3 DATA FLOW EXPLANATION

Data flow explanation simplifies the understanding of data movement and change in the system from input to output. Here, in this project, which seeks to convert a standard USB printer to a smart printer converted into wireless form using a Raspberry Pi, data flow is relevant in terms of how print jobs are carried out in a wireless system.

The central idea is to facilitate wireless print transmission across a local network with the help of a Wi-Fi router, a Raspberry Pi, and a printer using a USB cable. The operation starts with data transmission when there is a print request made from the user's device, either a laptop or a mobile phone. Both devices are already part of the same wireless network provided by the router. When the user requests the print, the details of the document are sent via the Wi-Fi network to the Raspberry Pi, which serves as an intermediary print server.

The Raspberry Pi does this request in a lightweight Linux OS that has pre-installed CUPS (Common UNIX Printing System). CUPS is the one that puts them in a queue, manages printing jobs, transforms them into printer-friendly ones, and then sends them via USB to the attached legacy printer.

Data flow can be broadly classified into the following stages:

1. User Device (Mobile/Laptop)

The process starts from the user choosing a document or file to be printed. The file is then transferred through Wi-Fi to the local network where the Raspberry Pi is located. Handheld devices such as smartphones, tablets, and laptops with Wi-Fi functionality act as the source of input of print commands. Native printing capabilities or third-party applications may be utilized based on the settings and operating system of the device.

2. Wi-Fi Router (LAN Gateway)

The router is the communication hub. It places the user's device and the Raspberry Pi on one network and in a position to share data packets. The router performs the role of sending and receiving data using TCP/IP protocols. This does not need the internet connection but a local network with a ecure connection.

3. Raspberry Pi (Wireless Print Server)

When print information arrives at the Raspberry Pi, actual processing is started. Raspberry Pi is a Linux operating system and has CUPS software support. CUPS translates received data, makes it compatible into a form that the printer can read (PDF, PostScript, PCL, etc.) and puts it in the job queue. Raspberry Pi completes the wireless communication and USB interface deficiency since it serves as a print server. Data transmission in the Pi occurs as follows: it reads the print job, sends print parameters, handles priorities of jobs, and then transmits the data to the USB port.

4. USB Printer (Output Device)

Connected to the Raspberry Pi itself through a USB cable, the conventional printer is streamed the printed document and starts printing. Due to the configuration of the Raspberry Pi, the printer is treated as if wireless. People no longer have to physically plug their devices to the printer.

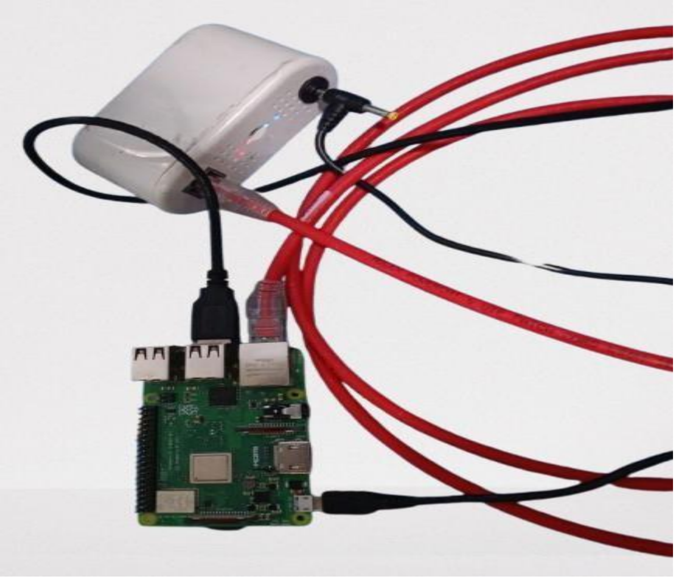

Figure 3.4 Proposed System Setup

The suggested Smart Printing System aims to transform an ordinary USB printer, say HP LaserJet P1108, into a wireless printer through the use of a Raspberry Pi 3. The cost-effective and space-friendly configuration makes it possible for the user to send remote print jobs from laptops, smartphones, or tablets within the same Wi-Fi network. The Raspberry Pi 3 functions as the processor, with a reduced Linux operating system and the CUPS (Common Unix Printing System) software installed, operating as a dedicated print server.

CHAPTER 4

4. HARDWARE COMPONENTS

4.1 RASPBERRY PI

Raspberry Pi 3 is a computer that sits on the level of a credit card as well as the brain of the intelligent printing system. It has Wi-Fi, Bluetooth, USB, HDMI, and microSD card slot. It is employed in the communication among the USB printer and the receiving of Wi-Fi print commands from laptops or cell phones.

In this project, the Raspberry Pi 3 executes software such as CUPS to make the wired printer a wireless printer. It connects to the network via Wi-Fi or Ethernet and sends print instructions to the HP LaserJet printer plugged into it. It is small, power-conscious, and appropriate for light tasks such as wireless printing

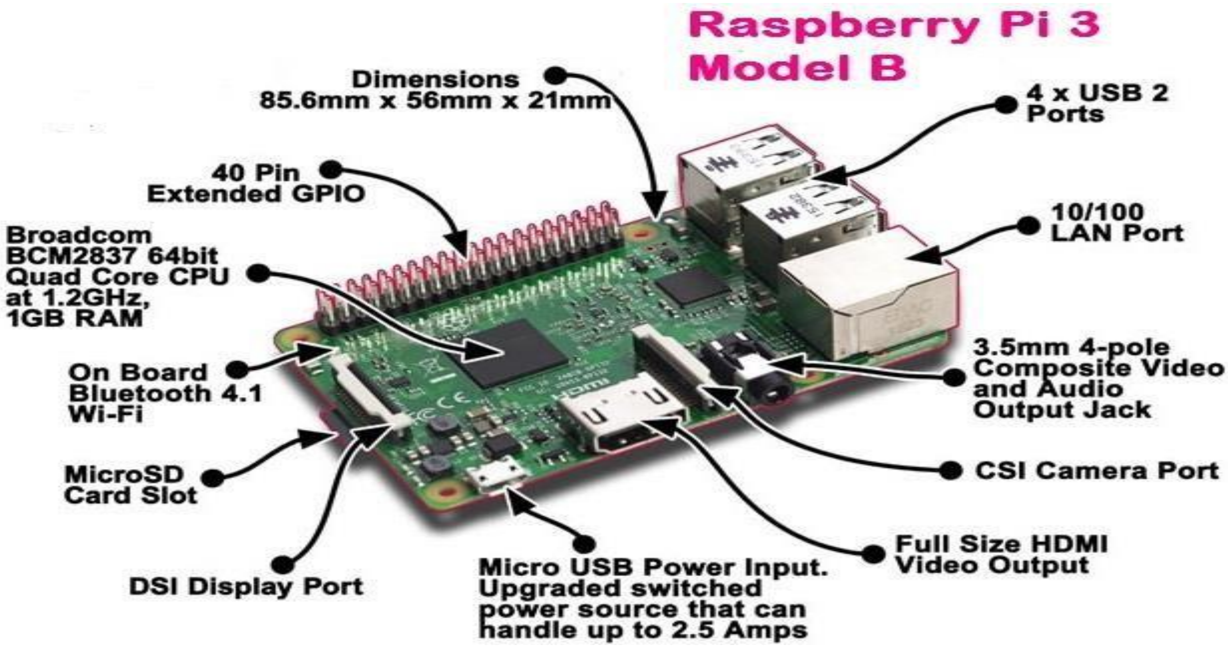

Figure 4.1 Raspberry Pi 3 Model B

4.2 USB PRINTERS

A USB printer is an external peripheral hardware that is attached to a host microcontroller or computer through a USB interface. It is one of the most utilized and stable printing media for general use, especially for personal use as well as official use. USB printers are stable and fast in connectivity, allowing rapid communication between the system and the printer without necessarily requiring a network.

It is here where the USB printer becomes an important component. The USB printer receives the print data from the Raspberry Pi directly via the USB cable. After processing the user's print request (which is received wirelessly), the Raspberry Pi transfers the file to the USB printer via the CUPS (Common UNIX Printing System) backend. This setup allows an ordinarily cabled printer to be utilized as a wireless-capable device, providing users with convenience and flexibility without requiring them to replace existing hardware.

USB printers are also highly compatible with any operating system and need minimal or no driver installation. They are plug-and-play devices and can transfer data at a quick speed, thus are ideal for high-volume or high-quality printing. As the connection is with USB, users can bypass the network setup and configuration, thus this is a cost-saving solution as well as easy to use.

Examples of USB Printers:

- HP LaserJet P1108
- Canon LBP2900
- Brother HL-L2321D
- Epson EcoTank L3211
- Samsung ML-1676
- Xerox Phaser 3020
- Ricoh SP 111

Figure 4.2 USB Printer Connected to Raspberry Pi

4.3 POWER SUPPLY & PERIPHERALS

A reliable power supply plays a crucial role in ensuring uninterrupted operation of the smart printing system. The Raspberry Pi, being the core component of the setup, must be continuously powered to manage printing tasks efficiently. Typically, the Raspberry Pi 3 requires a stable 5V/2.5A power input through a micro-USB connector. This consistent power flow is essential not only for the board itself but also for any peripheral devices—such as USB printers or Wi-Fi dongles—connected to it.

Since the Raspberry Pi is expected to function as a print server running software like CUPS

(Common Unix Printing System) and Samba services continuously, fluctuations or interruptions in power can lead to performance issues. These may include delays in processing print jobs, system crashes, or even data loss during transmission.

To mitigate these risks, using a high-quality power adapter with voltage regulation is recommended. Additionally, in settings where power cuts are frequent, an uninterruptible power supply (UPS) or a power bank with pass-through charging can be integrated to maintain continuous operation. Proper cable management and shielding may also help reduce electrical noise, ensuring smooth communication between the Raspberry Pi and peripherals.

This robust power management strategy ensures the smart print server remains active and dependable in environments like homes, schools, and small offices, where reliability is key for productivity.

4.4 NETWORK SETUP (WI-FI ROUTER)

Establishing a proper network configuration is essential to ensure seamless communication between the Raspberry Pi-based print server and the user devices, such as laptops, tablets, or smartphones. In this project, a Wi-Fi router is used to create a local area network (LAN) through wireless connectivity, allowing all devices to be interconnected even in the absence of an active internet connection.

The selected router—in this case, a Cofe Wi-Fi router—acts as the central hub for all network traffic. It enables the Raspberry Pi and client devices to communicate over the same wireless network, facilitating the transfer of print jobs to the server efficiently. This setup is particularly suitable for office or home environments where stable and secure communication is vital.

By assigning IP addresses dynamically using DHCP (Dynamic Host Configuration Protocol), the router ensures that each device on the network, including the Raspberry Pi, is easily discoverable and accessible. Once connected, users can send documents wirelessly to the print server using CUPS or network file sharing via Samba.

Moreover, the router’s encryption features (such as WPA2 or WPA3) add an additional layer of security, preventing unauthorized access and ensuring that sensitive data remains protected during transmission. The ease of setup and compatibility with a wide range of devices make this network configuration both practical and scalable for future enhancements.

4.5 LAPTOP

The laptop serves as one of the primary user-end devices for initiating print commands in the smart wireless printing system. It connects to the same wireless network as the Raspberry Pi and communicates with the print server either through a web interface provided by CUPS (Common Unix Printing System) or via network sharing protocols such as Samba.

Figure 4.3 Laptop (DELL)

Since most modern laptops come with built-in Wi-Fi capabilities, no additional hardware is required to connect to the wireless network. Once the connection is established, users can easily discover the Raspberry Pi print server, install the printer configuration (if necessary), and begin sending print jobs from applications like Word, PDF viewers, or web browsers.

The laptop acts as a user-friendly interface, allowing individuals to preview documents, modify print settings such as layout, page range, or duplex printing, and monitor the status of their print jobs. In a shared environment, multiple laptops can access the same printer concurrently, increasing productivity and eliminating the need for physically moving files or using USB drives.

Moreover, being portable, laptops enable users to print wirelessly from anywhere within the range of the Wi-Fi router, thus maximizing the convenience and flexibility of the smart printing system.

.

CHAPTER-5

5. SOFTWARE

5.1 RASPBERRY PI OS SETUP

As the initial critical step to start converting a typical USB printer into an intellectual wireless printer, placing a good and compatible operating system within the Raspberry Pi is required. The Raspberry Pi OS, being Linux Debian-based, is lightweight and tweaked specifically for Raspberry Pi hardware. An installation of this nature offers the necessary environment whereby print server software like CUPS and Samba can be installed and executed. By loading the OS onto a microSD card, having necessary peripherals plugged in, and setting the initial settings, the Raspberry Pi is now set to act as the wireless printing system's brain.

5.1.1 Steps for Setting Up Raspberry Pi OS:

- **Download Raspberry Pi Imager** from the official website https://www.raspberrypi.com/software/
- **Insert a microSD card** (at least 16GB) into your card reader.
- Use **Raspberry Pi Imager** to flash **Raspberry Pi OS** onto the card.
- Insert the card into the Raspberry Pi and connect peripherals (monitor, keyboard, mouse) or enable SSH for headless setup.
- Power on the Pi and perform the **initial setup**: type "**sudo raspi-config**" then Enable SSH and interfaces, Configure Wi-Fi
- **Update the system packages**: "s**udo apt update && sudo apt upgrade –y** "

This OS setup prepares the Raspberry Pi to act as a reliable and responsive print server on the local network.

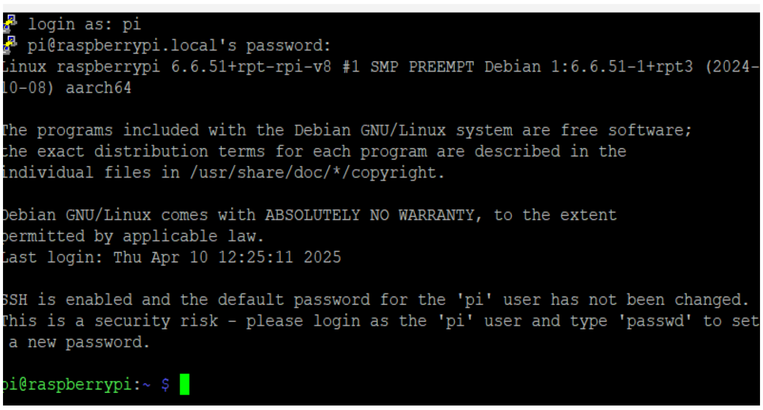

Figure 5.1 Raspberry Pi OS Boot Screen

5.2 **INSTALLING & CONFIGURING CUPS**

CUPS (Common UNIX Printing System) is an open printing system for Unix-like operating systems that allows a computer to act as a print server. With Raspberry Pi installation and configuration of CUPS, the device can accept and process print jobs and share the USB printer's access in the local network. CUPS is compatible with the Internet Printing Protocol (IPP) and integrates well with most printers and client devices like Windows, macOS, and Linux operating systems.

Raspberry Pi is an internet-enabled print manager with CUPS installed where users are able to remotely send print jobs. CUPS web interface is a simple means of accessing printer settings, monitoring print queues, and specifying access permissions.

1.a.1 **Steps to Install and Configure CUPS:**

- Install CUPS: "**sudo apt install cups -y**"

- Add user pi (or your current user) to the lpadmin group to manage printers:

"**sudo usermod -aG lpadmin pi**"

- Restart the CUPS service:"**sudo systemctl restart cups**"
- **Access the CUPS Web Interface:** Open a browser on any device connected to the same network and visit: http://<raspberrypi_ip>:631

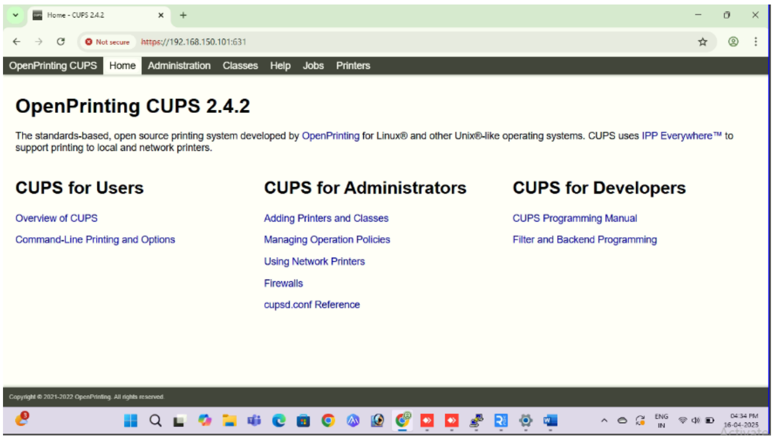

Figure 5.2 CUPS Web Interface

Through this interface, users can add printers, configure drivers, set default options, and manage print queues. This setup is crucial for enabling remote and wireless printing functionality.

5.3 **SETTING UP SAMBA**

Samba is a collection of free software that provides printer and file sharing across Linux PCs and Windows PCs. In smart wireless printing deployment, Samba is set up on the Raspberry Pi so that other devices on a network or Windows PCs will recognize the USB-plugged printer automatically.

Samba is a middleman that enables non-Linux computers to talk to the Raspberry Pi just as Windows machines are able to communicate with it as a file and print server. It enables users to just send print jobs from their own computers or desktop without needing the installation of particular drivers or command-line interfaces.

5.3.1 Steps to Install and Configure Samba:

- Install Samba: "**sudo apt install samba samba-common-bin -y**"
- Open the Samba configuration file: "**sudo nano /etc/samba/smb.conf**"
- Add the following configuration at the end of the file to share the printer:

"**[printers]**

comment = All Printers

path = /var/spool/samba

browseable = yes

guest ok = yes

writable = no

printable = yes

create mode = 0700"

- Restart the Samba service to apply changes: "**sudo systemctl restart smbd**"

5.4 **PRINTER CONFIGURATION VIA CUPS**

Common UNIX Printing System or CUPS is a Linux system for printing management and print queuing. Having installed CUPS on the Raspberry Pi, printer configuration is needed to enable wireless printing capability. This setup would make the USB printer become discoverable, setup correctly, and accessible through the network.

The users can install CUPS printers using a web interface so that they can easily add, delete, and manage printers.

5.4.1 Steps to Configure a USB Printer using CUPS:

- **Access the CUPS Web Interface** Open a browser on any device connected to the
- same network and enter:"**http://<raspberrypi_ip>:631"**
- In the CUPS web interface, go to **Administration** > **Add Printer**.
- Log in with the Raspberry Pi credentials.
- Select your connected USB printer (e.g., *HP LaserJet P1108*).
- Choose the appropriate driver (usually auto-selected)
- Set printer name, location, and description and select "share this printer"
- Finish setup and print a test page.

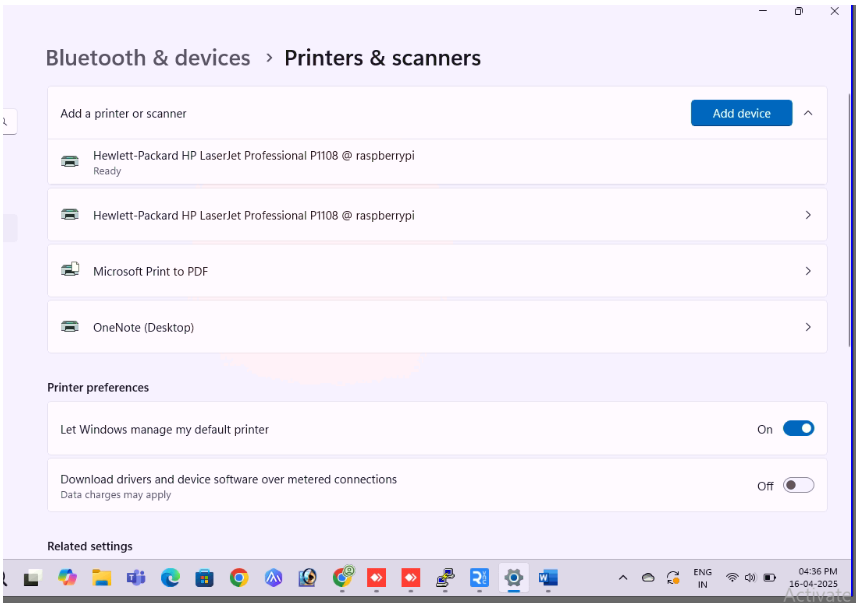

Figure 5.3 Printer Detected

CUPS makes it simple to manage printers from anywhere on the network. With this configuration, any device connected to the router can send print jobs wirelessly through the Raspberry Pi.

5.5 REMOTE PRINT MONITORING

Remote print monitoring is a built-in feature in any smart printing system, especially when the print server has a Raspberry Pi. It allows administrators and users to be able to watch and manage print jobs on the network from where they are situated on the network. For CUPS, there is an alternative in the form of its own web-based user interface giving real-time reports of the print queue, successful output, failures, and printer state.

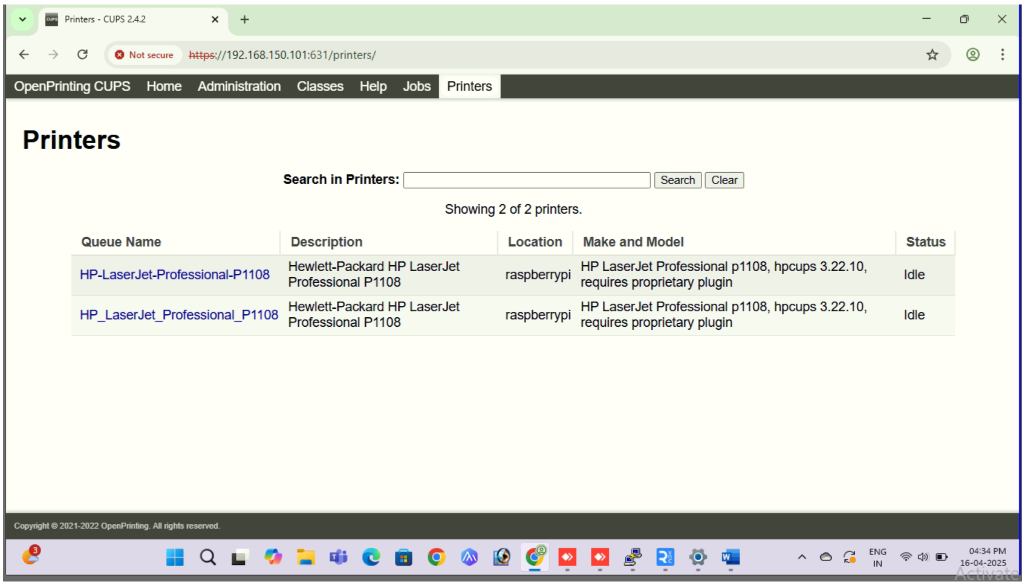

Figure 5.4 Printer Configuration Options in CUPS

Remote monitoring of print jobs provides greater control, prevents print queuing, and permits remote diagnosis without having to touch the printer or Raspberry Pi.

CHAPTER-6

6. **WORKING & RESULTS**

This section explains how the system operates functionally and prove the results obtained by turning an ordinary USB printer into a wireless printer device through Raspberry Pi. The implementation is in sequential order, yielding efficient remote printing on Wi-Fi devices.

6.1 **SETUP & BOOT**

Wireless printing with Raspberry Pi started from setting up and installing the hardware and software elements. The section indicates what was done to prepare Raspberry Pi to serve as a wireless print server and discusses how the boot process helps in ensuring continuity and reliability of service. The process was split into two broad stages: setup and confirmation of boot, both of which were important to facilitate USB printer functionality via Wi-Fi.

6.1.1 **Installation Procedure**

The procedure for installation entailed installing Raspberry Pi with a proper operating system, installing print services required, and connecting the USB printer. All these were carried out one after the other to avoid making any errors in the process of installation.

- Installing the Operating System

  Firstly, a microSD card was formatted, and official Raspberry Pi OS was installed. OS gave the appropriate environment to execute printer sharing services. Once the write process of OS image into SD card (official imager or same process) was done, it was installed into the Raspberry Pi and booted. The instant the Raspberry Pi started its initial boot; the default setup booted the system.

- Enabling Network Connectivity

  For remote access and wireless functionality to be enabled, the Raspberry Pi would have had to be connected to the local network. For this project, Ethernet was utilized to enable a constant internet connection. This allowed the Raspberry Pi to download packages it needed and be accessed via the local area network (LAN).

- Installing and Configuring CUPS

  After the network was set up, CUPS was installed from the terminal using the following command: "**sudo apt-get install cups"**

  CUPS is an open-source print system that enables network devices to print jobs to the Raspberry Pi. It also features a user-friendly web interface for printer and queue management. CUPS was installed, enabled, and configured to enable access from other network devices during installation. The configuration file was edited to grant users permissions and share printers.

- Port Configuration

  CUPS utilizes port 631 for its web interface. The port had to be opened and tested to verify that the users would be able to access the CUPS dashboard via their browser using the IP address of the Raspberry Pi (e.g., http://192.168.1.10:631). Once verified, this allowed administrators to manage printer settings and view print jobs from any connected device.

- USB Printer Detection and Configuration

USB printer was subsequently plugged in to the Raspberry Pi. The plugged-in printer was detected by CUPS and shown in the dashboard under "Add Printer." An appropriate printer from the current devices was picked, and a proper PPD (PostScript Printer Description) file was transferred for the chosen printer model. The file carries the proper driver functionality and further ensures that formatting of the print job is adequately taken care of.

Upon successful installation, the printer was successfully installed, properly named, and designated as the default printer. The installation made it simple for users to use it from any device on the network without having to install individual drivers on each device.

6.1.2 **Boot Verification**

There were tests performed after the first install on whether the system would boot after restarting. This meant shutting down and booting up the Raspberry Pi and testing whether the printer and print server would boot up normally.

- Service Initialization

Upon booting the Raspberry Pi, the Ethernet network connection re-established automatically. The CUPS service started automatically upon booting the system. This was confirmed by running the following command:"**sudo systemctl status cups** " . The output confirmed CUPS was running and in the background. Additionally, the printer remained an available device and all the settings were carried over from the previous session. This confirmed configuration files were saved and the system wasn't reconfigured after a power cycle.

- Printer Availability and Remote Access

To confirm the system was available: The CUPS web interface was accessed again with a laptop browser.

The printer chosen was displayed and read as "Idle – Ready to print." The printer started by printing out a test page in the way that it can accept print jobs.

All reverted to normal usage, which is that the setup was persistent, self-configured, and solid. This is the normal usage in real usage when the users prefer to have the printer online all the time without manual setup after power cycles or reboots.

- Network Discovery

Raspberry Pi was accessed using its local IP address, which was retrieved by running: **"hostname -I".** It was possible to print and see by other computers on the same network using this local IP. It may be possible to have a static IP if access was required, but in this instance, a dynamic IP was fine if it was known.

The entire installation and booting were completed successfully. Raspberry Pi served as a wireless print server correctly following a configuration session. Steps included installing OS, printing connectivity, CUPS installation and configuration, network sharing, and persistence of the test service.

These initial efforts allowed the subsequent tests to be performed successfully, including device connectivity and handling of print jobs, discussed in the subsequent sections.

6.2 DEVICE CONNECTIVITY

Upon completion of the Raspberry Pi and USB printer setup as a wireless print server, the next stage was to test which other devices could connect and if they could print to the configured USB printer. The ability for devices to connect to the Raspberry Pi is critically important for remote printing and is the driving factor in deciding how usable the system will be in real-world environments like homes, colleges, and offices.This section outlines what devices were used for connectivity such as laptops and mobile phones, how the printer was discovered via the network, and how communication was successfully achieved between client devices and the print server.

6.2.1 Setting up the Network Environment

For this project, the Raspberry Pi was connected to the local network via an Ethernet cable, allowing for steady communication and internet access. The remaining devices (laptops, smartphones) were connected to the same Wi-Fi network provided by the wireless router, creating a Local Area Network (LAN), allowing for communication between the Raspberry Pi and client devices.

Once the Raspberry Pi was connected and running, the IP address was revealed by issuing the command: "**hostname -I**".

The IP address (e.g., 192.168.1.10) was used by other devices to access the CUPS web interface by entering: "**http:// <raspberry pi ip>:631**" into a web browser. This created a means to access the print server remotely from any device on the same network, without a wired connection.

6.2.2 Connecting to the Printer with Laptops and Desktops

The following steps were arrived at using a laptop or desktop plugged into the same network:

- opened the browser and reached the CUPS webpage using IP:PORT 631
- Your installed printer will appear on the homepage with the status of “Idle – Ready to print”.
- Allow user to see printer configuration features, status of print queue, jobs, etc.
- Using the add printer feature of CUPS, the user was able to map the Raspberry Pi printer as a network printer to their computer.

No drivers or plug-ins need to be installed on the client side, as all formatting and conversions are done at the printer side by CUPS. This creates a single point menu for the user and removing the user complexity, and their works on the setup was really user-friendly.

Since it was getting through the tests for the connectivity handling part, I confirm some stuff although the Raspberry Pi-based print server seemed available on more than a few entities via the same system. Laptops and smartphones would find and use the configured printer without the need to do arcane driver installations and manual configurations. Confirmed successful connectivity, the system works in a real-world (this global) environment without additional extensive setup effort.

6.3 PRINT COMMAND TESTING

After successfully connecting the devices, it was time to test the actual printing from different devices. In this phase, our goal was to test after setting up a Rasberry Pi as a print server with CUPS, if the Raspberry Pi was able to receive print jobs from laptops and mobile phones correctly and process it properly. This section describes how print commands are issued, what formats were tested, how the system responds with output, and general print handling behavior.

6.3.1 Print Testing by Laptop

To test, a laptop connected on the same network to the raspberry pi open the web interface of cups with http://:631

Once on the dashboard, the default printer was visible. The following steps were carried out:

- A test .txt file you selected and uploaded by clicking the "Print Test Page" or the "Print File" on the CUPS interface.
- For example, with that setup the print job would be immediately attached to the print queue.
- We then sent the job to the printer and printed the contents of the text file.

A built-in TIHS feature even enabled the setting of job priority and several print options like number of copies, page size and orientation. Fortunately, all had been correctly handled by the Raspberry Pi and interpreted as expected by the printer.

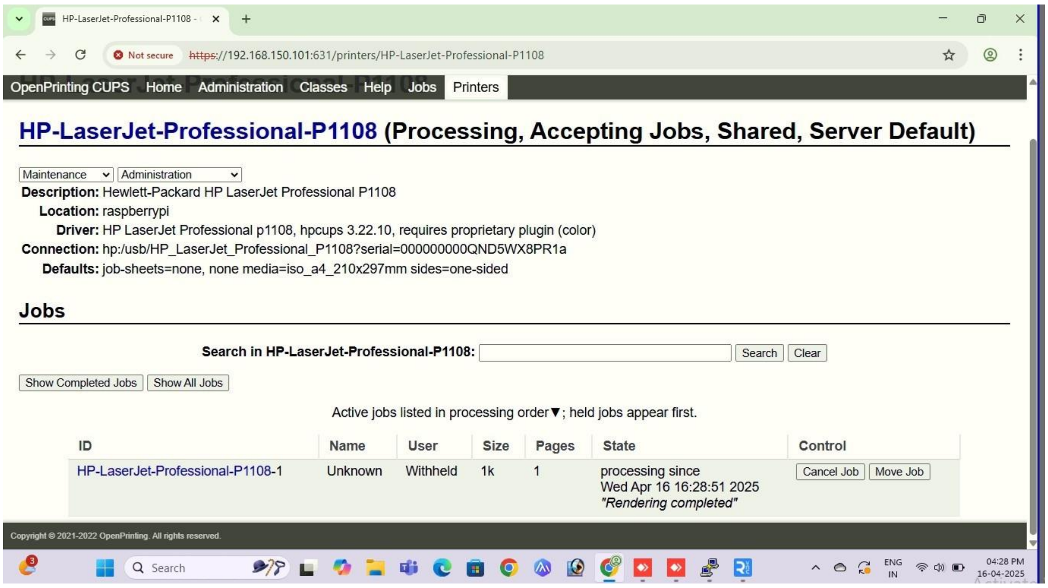

Figure 6.1 Wireless Printing Test from Windows

6.3.2 Print Testing via Mobile Device

Second, we accessed the CUPS interface using a smartphone on the web browser. Test steps were:

- Posting a test .pdf file with UI
- Directly tell the phone to print.
- Noticing that job appeared in CUPS queue as expected and a few second later was being processed by printer.

The test confirmed that with no need to install any third party print app, mobile users can interact with Raspberry Pi print server via browser. Increased portability and convenience of the system especially for home & small office deployments. with the Raspberry Pi print server simply through a browser interface. It enhanced the portability and convenience of the system, especially in home and small office setups.

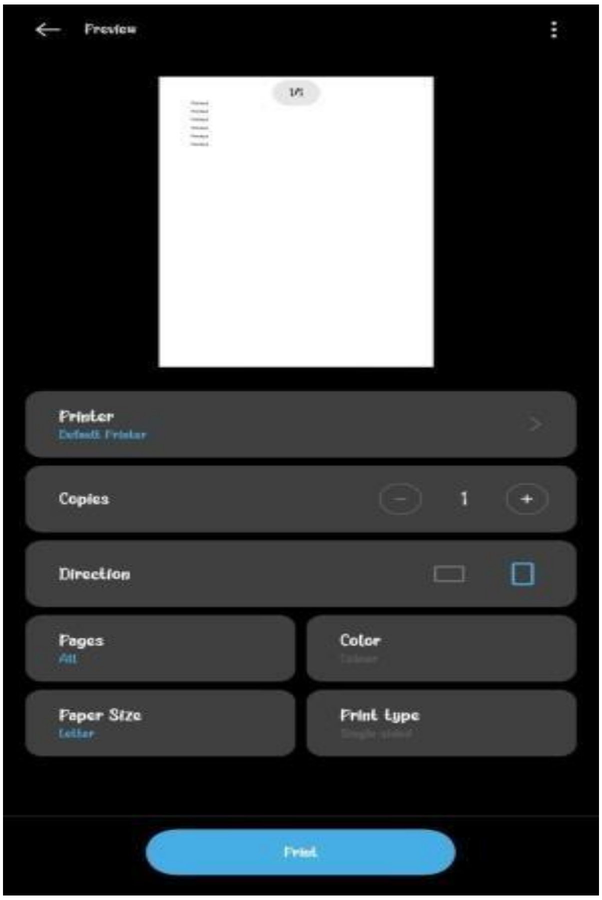

Figure 6.2 Wireless Printing Test from Mobile

6.3.3 System Feedback & Job Monitoring

For print testing, the CUPS interface offered print status in realtime as well

- "Processing — printing page 1 of 1"
- Idle Ready to print
- "Completed at [time]"

Moreover, various system CUPS kept log for each job like jobId, user name, filename and status. This gave us a nice audit history of print activity for monitoring and debugging.

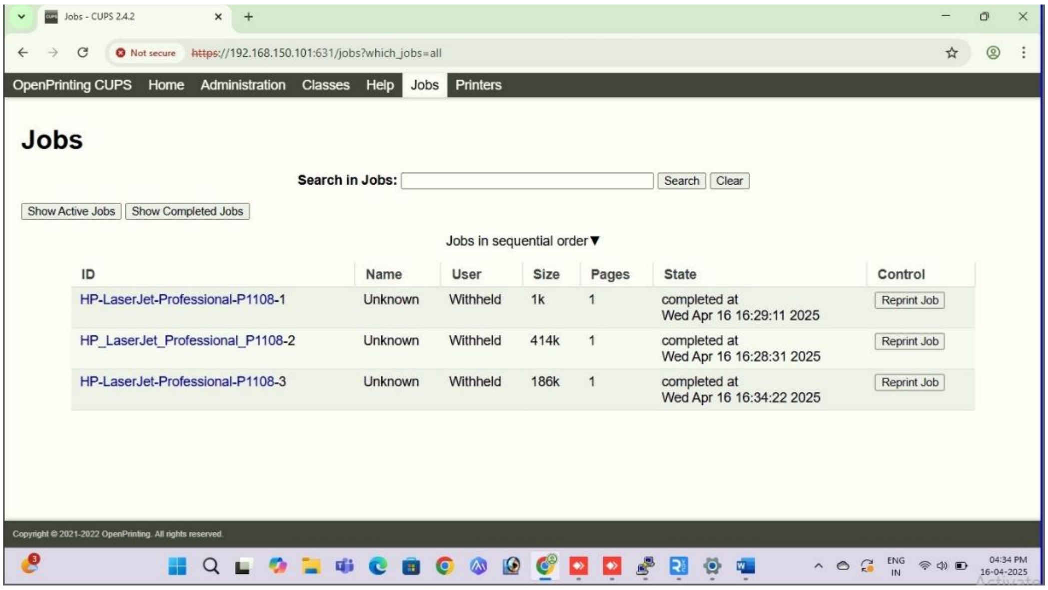

Figure 6.3 Terminal Output Showing Print Jobs

6.3.4 Performance and Observations

- Stall Time: Sending a job to receive print output was noted to stall for approximately 3–5 seconds when text and image files
- Data Loading: If in case the submitted jobs were submitted consecutively, they will be loaded into the queue and get executed in sequence.
- Error Status: For example, a job rejected due to corrupt file on its own that system would have thrown an error message further on CUPS dashboard.

The above tests proved that the system is responsive/working, accurate and capable for common print tasks from different device type without any specific printer configuration on client end.

The testing of print command confirmed that Raspberry Pi-based wireless print server works for basic cases at which the integration was implemented. It had happily picked up and processed print jobs on laptops and smartphones from a range of file formats over the network. The CUPS interface was of great help in terms of implementing a simple and usable solution to manage as well monitor print jobs and hence streamline the whole process.

Figure 6.4 Successfully Printed Document

6.4 SUCCESFUL WIRELESS PRINTING

The ultimate goal of this project was to convert a common USB printer into a wireless printer using a Raspberry Pi. After successfully performing system configurations, connectivity tests, and printing command testing, the system was subjected to an evaluation regarding real-time functionality and efficiency under normal working conditions. This section presents the results and observations demonstrating the successful execution of wireless printing.

6.4.1 Real-Time Printing Across Multiple Devices

With CUPS, the Raspberry Pi was set up as a print server so that print commands could be sent over Wi-Fi from various devices, including laptops and smartphones. The key attraction of this setup was that users were free to print from anywhere within network range without having to manually connect to the printer.

Tested Use Cases:

- Printing from a laptop using the web interface of CUPS.
- Uploading documents from the mobile browser to the print queue.
- Sending multiple print jobs simultaneously from different devices.

In all instances, the system got the job, processed it, and printed the output successfully

6.4.2 Integration With Web Interface

The web interface of CUPS was instrumental in soliciting wireless printing. The following features were offered:

- Visual display of printer status (Idle, Printing, Completed).
- Uploading documents directly for printing.

- Management of queued jobs (for example, to cancel, pause, or reprint).

This made it possible for normal users, having little or no technical background, to access and make use of the printing service without any knowledge of command-line use or system-level access.

6.5 **TROUBLESHOOTING & FIXES**

While the overall system performance after configuration was as expected, there were encountered several real-time problems in testing that required systematic troubleshooting and correction. This section introduces an exhaustive description of the challenges and remedial steps taken to identify them. These experiences contributed much towards knowing the practical limitations of embedded systems and the application portability of the wireless printing setup.

6.5.1 Printer Not Detected by CUPS

One of the initial challenges during implementation was that the CUPS interface failed to detect the printer attached through USB. After physically connecting the printer to Raspberry Pi, here was the expected behavior-the device would show up in the CUPS web interface list of available printers as soon as it was done. This, however, did not happen at first. To find out what it was, some troubleshooting was done using the lsusb command on the terminal, which lists all USB devices connected to the system.

It confirmed that the printer was recognized as hardware; CUPS, however, would not register it. The cause- identified delayed service startup-CUPS daemon started before the printer was fully initialized. Solution: unplugging and re-plugging in after Raspberry Pi had booted then a restart of Cups service using sudo systemctl restart cups. After refreshing the browser interface, the printer showed up in the list, and normal configuration could proceed.

6.5.2 Web Interface Not Accessible

Another issue was that the local network had inability to access CUPS interface through the web. The conventional URL address used to access the interface**(http://<raspberrypi_ip>:631)** sometimes refused to load. This was indicative of either the CUPS service not running or the network port (631) being blocked or misconfigured. To check this, the status of the system using,

the command **sudo systemctl status cups** was used. If there was a case where the service had failed or not started at boot, this was overcome by restarting the service using **sudo systemctl start cups**.

The firewall was as well checked to ascertain whether the port 631 is being blocked. Since no firewall was configured at custom, this was not a valid issue. At times, caching or DNS problem could lead to failure in page loading. Switch to incognito tab or clear history and refresh the DNS cache on the browser. CUPS operation and accessibility in a local area network have thusn't been confirmed.

6.5.3 Incorrect Driver or PPD File Mismatch

Another critical issue was associated with either using improper or missing PPD files. When the printer was added through the CUPS web interface, the user was asked to select the appropriate PPD file that described the capabilities of the respective printer. In one case, a wrong driver was chosen and it produced garbled or incomplete printouts. Sometimes, it would not print at all or would break into fragments. Thus, printing became possible after consulting both the printer's manual and the manufacturer's website, to identify a suitable PPD file.

The practical troubleshooting phase was one of the highly significant factors for learning during the real deployment and maintenance of the system. Hardware detection, driver mismatches, services outage, and network access were issues that had to be hands-on and practical. On the performance end, stabilization and optimization of the system were achieved through command-line diagnostics, log file analysis, and some basic administrative actions. This phase further enrichened the technical knowledge of managing embedded Linux systems with the importance of foresight in designing-andplanning-for real-world variability. Ultimately, the entire written proceedings rendered the wireless printing system dependable, scalable, and user-friendly: ready for sustained use under conditions of everyday application.

6.6 **OUTPUTS**

After setting up and testing the Raspberry Pi-based wireless printing system, output types were produced to examine its functional correctness and practical usability. Digital indications, such as the status read from a CUPS web interface and terminal responses, were some outputs as were the physical outputs, consisting of printed documents from a USB printer. This section depicts and summarizes the key results recorded during the implementation with indications of the various use scenarios to prove that the system is functioning as was intended.

6.6.1 Output for CUPS Web Interface

The CUPS web interface allowed the user to manage and monitor the printer from remote devices. When accessed via the browser (http://:631 richly printer-related info would be displayed.

- Printer name and model.
- Connection type (USB).
- Status messages (e.g., Idle, Printing, Completed).
- Print job history and queue.

One of the most significant outputs would have been the live status update during job execution. Once a user issued the print job, CUPS would farther have displayed the message "Processing – Printing page 1 of 1" and automatically switched it to "Idle – Ready to Print" when the job was completed. Other functions like administrative ones would also be present, such as Pausing or killing a job, Adding new printers, or Editing the settings of existing printers.

6.6.2 USB Printer Printed Output

The principal output is the actual printed document that has been printed by the USB printer after a job from a connected device is sent to it. The printing operation has worked with various formats, namely:

- Text Files (.txt)
- PDF Documents (.pdf)
- Image Files (.png, .jpg)

Cleanly as well printed, the documents came out well aligned and true to the respective digital files sent through the CUPS interface. The print job sent from the laptop or mobile device was well received and printed successfully.

Printed Output Observations :

- Excellent font and image clarity were maintained in the printout.
- Page size, orientation, and margins were as per configuration.
- No distortion or data loss suffered by the prints.

The outcome, however, actually proved that the smart printing system is working perfectly. All the dimensions of the solution have brought in outputs that are clear and possible to act on, from the dashboard-based GUI to the command-line logs and real-time job status to the actual printout. If there was anything, that would have evidentially proved the robustness of the solution, - the generated outputs emergent across different devices, a clean, user-friendly interface, and the activity log. Jointly, these outputs prove that indeed a Raspberry Pi can give to any conventional USB printer the same wireless printing functionality as if it were a completely new system.

6.7 OUTPUT ANALYSIS

Assessment results from outputs during the testing and deployment phases of the wireless printing system based on Raspberry Pi give enormous insight into practical feasibility, efficiency, and overall success of the project. Every aspect from the onset of sending out a print job to the termination of physical dissemination of documents was keenly observed for performance evaluation in different real-time scenarios.

Quality of Physical Print Output

The most significant output of this system was the actual printout produced from the USB printer. It was the immediate evidence that the Raspberry Pi behaved as a fully functional print server. . Numerous file formats such as .txt, .pdf, .png, and .jpg were used during testing, all being printed without ambiguity, proper formatting, and complete data retention. Successful results verified that the CUPS (Common Unix Printing System) on Raspberry Pi did precisely process and render documents and that the PPD being used for the printer was accurate and compatible.

User Interface Feedback (CUPS Dashboard)

The space beyond the prints shows how the CUPS web interface plays an even bigger role in user interactions with the system as a visual confirmation. With every new job that is submitted, queued, or printed, the real-time statuses of the jobs such as "Idle," "Printing," or "Completed," refreshes accordingly. Submit, cancel, reprint, and it manages job priorities are all enjoyed in the experience like one would have with commercial wireless printers. For a lay user, this added a lot of usability to the system.

Outputs from System-Level Terminals

Interface at the command line on Raspberry Pi also proved significant in checking output. Commands such as lpstat, lpq, and systemctl status cups provided the real-time system feedback on job status, printer readiness, and service health. This form of back end output became very important in confirming the operational situation of the system, particularly at boot, during updates, or troubleshooting sessions. The terminal became an added layer of transparency for system administration.

Print Speed and Responsiveness

Speed and responsiveness were two key output evaluation areas. The average time from job submission to the initiation of printing consistently remained between 3-5 seconds for small files and slightly longer for larger documents. The system continued to remain stable and responsive even in the face of multiple users submitting print requests in rapid succession, demonstrating efficient management on the part of the CUPS job queue and reliable real-time operations on the Raspberry Based Pi.

Device Compatibility and Multi-Platform Access

Cross-platform compatibility proved to be another major success criteria that came through analysis of the output. The system included devices running on Windows, Linux, and Android platforms without requiring any additional client-side drivers or software. Using Chrome or Firefox on a mobile device, the CUPS interface would be accessible to upload documents just as easily as a desktop or laptop. This wide compatibility made the system versatile and suitable for environments with mixed devices.

System Behavior after Reboot

This observation is about the practical high performance of output. The behavior of the system after reboots was consistent with all settings-from cup configuration to the user preference and default printer selection-retained by the Raspberry Pi through power cycles. The automatic startup of the print service ensured that the users do not have to redo any setup steps because the system was proven stable, persistent, and ready for continuous deployment in headless environments.

The final comprehensive analysis of outputs-from physical prints to visual dashboards, terminal feedback, and error logs-did confirm that the Raspberry Pi-based smart printing system worked exactly as it was supposed to. It was transforming digital input into real high-quality physical documents safely and accurately reacts quickly to user commands and remains in a constant operational state across devices and sessions. These outputs have confirmed the success of the project both technically and functionally and demonstrated its readiness for practical use in various real-world situations.

CHAPTER-7

7. CONCLUSION AND FUTURE SCOPE

This is to say that through this project, a normal USB printer has been converted into a smart-wired printer, taking advantage of Raspberry Pi and the open-source software tools-on CUPS. We wanted to minimize its need for wired systems and provide a really cheap solution for wireless printing considering home use, education, or small-size office installations. The users printed via a browser interface from any device connected over same Wi-Fi, with even some real-time tracking of job status, cross-platform compatibility, persistent service configuration, and good accessibility. The entire solution thus boasts great efficiency, reliability, and user-friendliness. The CUPS dashboard completed all printing tasks, while the Raspberry Pi being an inexpensive power-saving embedded system served as a bridge between the USB hardware and the contemporary demands of wireless solutions. It appears to meaningfully satisfy the functional requirements, therefore this option seems viable for implementation.

In the future, enhancements can be made to support popular voice assistants like Alexa or Google Assistant so that in cases like a "give command" for "Alexa, print my document" or "Hey Google, check printer status". On the further upgrade lists may be cloud printing options, enabling users to print documents from anywhere where Internet access to the Raspberry Pi is provided. Another web interface is an option for mobile uses or even a mobile app for even easier use or convenience. Enhancing system intelligence by integrating further functions like print history logs or scheduled printing or automatic error notifications would seem like a good idea. A common path should be created for user role management and multi-level access controls, where administrators can monitor print usage blocking according to user level access. Raspberry Pi hardware updates can be carried out that upgrade it on its own or replace it with similar low-power single-board computers for good performance. It will be quite robust if it integrates storage support to keep frequently printed documents or for print later. Going forward, as technology improves, the main framework developed through this project will serve as a solid base for building advanced, voice-enabled, cloud-connected, and AI-supported smart printing systems for the future.

CHAPTER-8

8. **REFERENCES**

[1] Manish Kumar Reddy Gangula, Pradeep Kumar Laghavarapu, Mogili Vamsi Krishna, Masapalli Abhishek, Munagala Venkatesh. "Wireless Printing Using IoT." *IRE Journals*, Volume 1, Issue 8, February 2018, pp. 56–58.

[2] P. Mahitha. "Wireless Printer Fastener Using Raspberry Pi." *IJSTE - International Journal of Science Technology & Engineering*, Volume 3, Issue 1, July 2016, pp. 154–158.

[3] A. Pandey, R. Mishra, S. Sharma. "A Review Paper on Wireless Printing Using IoT." *International Journal of Engineering Research & Technology (IJERT)*, Volume 6, Issue 11, November 2017, pp. 232–234.

[4] R. Patel, N. Shah. "IoT-Based Printing System for Remote Locations." *International Journal of Scientific and Research Publications*, Volume 8, Issue 2, February 2018, pp. 47–50.

[5] M. B. Bhise, A. V. Bhole. "Implementation of Print Server Using Raspberry Pi for Wi-Fi Printing." *International Journal of Research in Engineering, Science and Management*, Volume 4, Issue 6, June 2021, pp. 12–15.

[6] R. A. Jain, S. Patel, A. Chaudhari. "Smart Print System Using Raspberry Pi." *International Journal of Computer Science and Mobile Computing*, Volume 7, Issue 4, April 2018, pp. 133–137.

[7] A. Agarwal, R. Gupta. "Wireless Printing over LAN Using Raspberry Pi." *International Journal of Computer Applications*, Volume 172, Issue 8, August 2017, pp. 23–26.

[8] G. Choudhary, S. Singh. "Low Cost and Portable Wireless Print Server Using Raspberry Pi." *International Research Journal of Engineering and Technology (IRJET)*, Volume 5, Issue 3, March 2018, pp. 1452– 1456.

[9] B. R. Mulla, P. K. Patil. "Wi-Fi Printing Using Embedded Systems." *International Journal of Innovations in Engineering Research and Technology*, Volume 6, Issue 7, July 2019, pp. 99–102.

[10] A. R. Pujari, D. U. Khalkar. "Design of Smart Printer System Based on Raspberry Pi." *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Volume 6, Issue 3, March 2017, pp. 187–190.

[11] V. D. Patel, R. P. Vyas. "Wireless Communication with Printers Using Raspberry Pi." *International Journal of Engineering and Technical Research*, Volume 4, Issue 5, May 2016, pp. 47–49.

[12] P. R. Deshmukh, A. K. Yadav. "Internet Printing Protocol Implementation with Raspberry Pi." *International Journal of Emerging Technologies and Innovative Research*, Volume 6, Issue 2, February 2019, pp. 487–491.

[13] A. N. Kadam, S. S. Patil. "Portable Wireless Printing System Using Embedded Platform." *International Journal of Engineering Trends and Technology*, Volume 50, Issue 3, August 2017, pp. 170–174.

[14] M. S. Vali, R. D. Deshmukh. "Smart Office Solutions Using IoT: Wi-Fi Printing." *Journal of Advances in Science and Technology*, Volume 15, Issue 1, 2018, pp. 128–132.

[15] R. Kumar, A. Sharma. "CUPS-Based Smart Print Management Using Raspberry Pi." *International Journal of Scientific Engineering and Research (IJSER)*, Volume 6, Issue 9, September 2018, pp. 82–85.

[16] L. K. Sharma, N. K. Jaiswal. "Smart Print Server Deployment in Schools Using Raspberry Pi." *International Journal of Education and Management Engineering*, Volume 8, Issue 4, 2018, pp. 21–26.

[17] T. K. Reddy, K. P. Sharma. "Role of Embedded Linux in Print Server Implementation Using Raspberry Pi." *International Journal of Advanced Research in Computer and Communication Engineering*, Volume 5, Issue 12, December 2016, pp. 345–348.

[18]S. R. Verma, D. R. Patil. "Automated Printing System over WLAN Using Raspberry Pi." *International Journal of Engineering Research in Electronics and Communication Engineering*, Volume 5, Issue 11, November 2018, pp. 101–104.

[19] S. G. Bhoyar, A. S. Kasar. "Design of Wireless Printing Architecture Using Open Source Tools." *International Journal of Innovative Research in Computer and Communication Engineering*, Volume 7, Issue 3, March 2019, pp. 225–229.

[20] D. R. More, P. R. Patil. "A Survey on Wireless Printing and Print Servers Using IoT."

International Journal of Research and Analytical Reviews (IJRAR), Volume 5, Issue 4, October–December 2018, pp. 547–552.

APPENDIX

Install the OS on the SD Card

Open the putty software and login to your Pi using the creditionals we have used while installing the os

#Ping the Raspberry Pi

ping raspberrypi

 # Enable ssh Into Raspberry Pi

ssh pi@raspberrypi

 # Unblock Wi-Fi

rfkill list

sudo rfkill unblock wifi

 # Reboot Raspberry Pi

sudo reboot

 # Enable VNC & SSH

sudo raspi-config

Interface Options > VNC > Enable

 # Find Raspberry Pi IP Address

hostname –I

#Now go to RealVNC Viewer and type the ip address and connect to pi and check the wifi connection and connect the wifi to the Raspberry Pi

#Update & Upgrade Raspberry Pi Packages

sudo apt update

sudo apt upgrade

 # Install CUPS

sudo apt-get install cups

Add the pi user to the lpadmin group

sudo usermod -a -G lpadmin pi

 # Allow remote access for CUPS

sudo cupsctl --remote-any

 # Restart CUPS service

sudo /etc/init.d/cups restart

 # Check IP Configuration

ifconfig

 # Check the Access of CUPS Web Interface

http://192.168.150.103:631 (192.168.150.103 is our pi ip address)

 # Install Samba

sudo apt-get install samba
 # sudo nano /etc/samba/smb.conf

sudo nano /etc/samba/smb.conf

[printers]

guest ok = yes

read only = no

 # Restart Samba Service

sudo systemctl restart smbd

 # Install Some HP Printer Packages (if required) sudo apt install hplip hplip-gui printer-driver-hpcups sudo

hp-setup –i

 # Open CUPS Web Interface

http://192.168.150.103:631 (192.168.150.103 is our pi ip address)

 # Add Printer

Go to **Administration > Add Printer**

Select the printer and click **Start Sharing**

Select the correct printer driver from the list according to the printer

 # Add Printer on Laptop

On the laptop, go to printer settings and add the printer shared

 # Enable Print Service on mobile

Go to **Settings > Connected devices > Connection preferences > Printing**
Once enabled, your USB printer (shared via Raspberry Pi) should appear.
If it doesn't, tap **Add printer > Add by IP address**, and enter your Raspberry Pi's IP and port 631

 # Access the Print Option

Open a document, photo, or webpage, and tap the **Share** icon, then select **Print**.