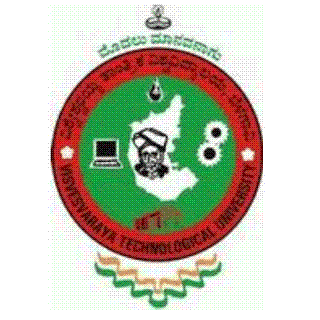
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI - 590 018**



PROJECT WORK PHASE-2

REPORT

on

**Biometric Based Payment System**

(Sponsored by Institute)

***Submitted in the partial fulfilment of the requirements for the award of the degree of***

**BACHELOR OF ENGINEERING**

**in**

**ELECTRONICS & COMMUNICATION**

**Project Associates**

**S SUHAS 4BD20EC060**

**SHREYAS P 4BD20EC066**

**VYSHNAVI H K  4BD20EC090**

**CHITRA A 4BD21EC408**

**Dr**. **BANUMATHI K L**

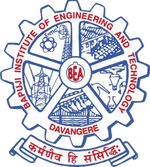
**M. Tech., Ph.D.**

**Project Guide**

**Dr. G.S. SUNITHA** **Dr. H. B. ARAVIND**

**M.Tech. (DEAC), Ph.D., MISTE, FIETE., FIE** **M.E., (Env. Engg.) Ph.D., FIE., MISTE., FIEE.**

**Program Coordinator Principal**

****

**Bapuji Educational Association®**

**Bapuji Institute of Engineering and Technology**

**Davangere-577 004**

**Department of Electronics & Communication Engineering**

**2023-2024**

**Institute Vision**

To be a centre of excellence recognized nationally and internationally, in distinctive areas of engineering education and research, based on a culture of innovation and invention.

**Institute Mission**

BIET contributes to the growth and development of its students by imparting a broad-based engineering education and empowering them to be successful in their chosen field by inculcating in them positive approach, leadership qualities and ethical values.

**Department Vision**

To be in the forefront in providing quality technical education and research in Electronics & Communication Engineering to produce skilled professionals to cater to the challenges of the society.

**Department Mission**

M1: To facilitate the students with profound technical knowledge through effective teaching learning process for a successful career.

M2: To impart quality education to strengthen students to meet the industry standards and face confidently the challenges in the programme.

M3: To develop the essence of innovation and research among students and faculty by providing infrastructure and a conducive environment.

M4: To inculcate the student community with ethical values, communication skills, leadership qualities, entrepreneurial skills, and lifelong learning to meet the societal needs.

**COURSE LEARNING OBJECTIVES:**

This course will enable us to:

* Prepare with foundation knowledge in a project domain through designing, and implementing.
* Introduce to various emerging fields in electronics and communication.
* Provide an opportunity to exercise the creative and innovative qualities, to become technopreneurs in group project environment.
* Have hands-on experience in the related field so that students can integrate and reinforce what has been taught in the classroom.

**ABSTRACT**

The increasing need for secure and convenient payment methods has fueled the exploration of biometric authentication systems. This project presents a Biometric Based Payment System designed to streamline transactions while prioritizing security. The system employs a fingerprint sensor (R307) to ensure robust user identification, complemented by a 4x4 keypad for versatile input and a 1.3” OLED (Organic Light Emitting Diode) display for clear feedback. At its core, the system leverages an ESP32 microcontroller and robust SHA-256 encryption to protect sensitive information during storage and transmission. Transaction data resides in a centralized MySQL database on an Apache server, and users manage payment history through a web interface. The system emphasizes security and ease of use, with the potential for future exploration of multi-factor authentication, offline capabilities, and blockchain integration to further enhance its robustness.

**ACKNOWLEDGEMENT**

We would like to acknowledge the help and encouragement given by various people during the course of this project.

We would like to express our sincere gratitude to our guide, Dr. Banumathi K L, Assistant Professor, Department of Electronics and Communication Engineering, Bapuji Institute of Engineering and Technology, Davangere, for her guidance, continuous support, and motivation in completing the project successfully.

We would also like to thank the project coordinators, Dr. Leela G H, Professor, and Mr. Yogesh K O, Assistant Professor, for their kind coordination during the course of the work.

Our sincere gratitude goes to Dr. G S Sunitha, Professor and Head of the Department of Electronics & Communication Engineering, BIET, Davangere, for her kind support, guidance, and encouragement throughout the course of this work.

We are thankful to our beloved principal, Dr. H B Aravind, for providing an excellent academic environment.

Additionally, we wish to thank our Institute for sponsoring our project.

We would also like to thank all the teaching and non-teaching staff of the Department of E&CE for their kind cooperation during the course of the work. The support provided by the college and departmental library is greatly acknowledged.

Finally, we are thankful to our parents and friends, who helped us in one way or another throughout our project work.

Project Associates

S Suhas

Shreyas P

Vyshnavi H K

Chitra A

**Table of Contents**

[CHAPTER 1 1](#_Toc167792386)

[1. INTRODUCTION 1](#_Toc167792387)

[CHAPTER 2 2](#_Toc167792388)

[2. LITERATURE SURVEY 2](#_Toc167792389)

[CHAPTER 3 5](#_Toc167792390)

[3. PROBLEM STATEMENT 5](#_Toc167792391)

[CHAPTER 4 6](#_Toc167792392)

[4. OBJECTIVES AND METHODOLOGY 6](#_Toc167792393)

[4.1 OBJECTIVES 6](#_Toc167792394)

[4.2 METHODOLOGY 7](#_Toc167792395)

[4.2.1 Block Diagram 7](#_Toc167792396)

[4.2.2 Flow Chart 8](#_Toc167792397)

[CHAPTER 5 10](#_Toc167792398)

[5. HARDWARE DESCRIPTION AND SOFTWARE DESCRIPTION 10](#_Toc167792399)

[5.1 HARDWARE REQUIREMENTS 10](#_Toc167792400)

[5.1.1 ESP32 10](#_Toc167792401)

[5.1.2 R307 fingerprint sensor module 11](#_Toc167792402)

[5.1.3 1.3" OLED displays 12](#_Toc167792403)

[5.1.4 4×4 matrix keypad 12](#_Toc167792404)

[5.1.5 Power Supply 13](#_Toc167792405)

[5.1.6 18650 li-ion battery 13](#_Toc167792406)

[5.2 HARDWARE INTERFACE 14](#_Toc167792407)

[5.2.1 R307 fingerprint sensor module interface with ESP32 14](#_Toc167792408)

[5.2.2 1.3” OLED display interface with ESP32 14](#_Toc167792409)

[5.2.3 4×4 matrix keypad interface with ESP32 14](#_Toc167792410)

[5.3 SCHEMATIC DIAGRAM 15](#_Toc167792411)

[5.4 SOFTWARE REQUIREMENTS 16](#_Toc167792412)

[5.4.1 Arduino IDE 16](#_Toc167792413)

[5.4.2 Hypertext Transfer Protocol (HTTP) 16](#_Toc167792414)

[5.4.3 Apache 17](#_Toc167792415)

[5.4.4 SQL (Structured Query Language) 17](#_Toc167792416)

[5.4.5 PHP (Hypertext Preprocessor) 17](#_Toc167792417)

[5.4.6 HTML (Hypertext Markup Language) 18](#_Toc167792418)

[5.4.7 JavaScript (JS) 18](#_Toc167792419)

[5.5 SOFTWARE FLOW 19](#_Toc167792420)

[CHAPTER 6 20](#_Toc167792421)

[6. RESULT AND DISCUSSION 20](#_Toc167792422)

[APPLICATIONS 21](#_Toc167792423)

[ADVANTAGES 22](#_Toc167792424)

[LIMITATIONS 23](#_Toc167792425)

[CONCLUSION 24](#_Toc167792426)

[FUTURE SCOPE 25](#_Toc167792427)

[REFERENCES 26](#_Toc167792428)

[APPENDIX A- Data Sheets 27](#_Toc167792429)

[APPENDIX B – Code 30](#_Toc167792430)

**Table of Figures**

[Figure 4.2.1.1: Block Diagram 7](#_Toc167801868)

[Figure 4.2.2.1: Enrolment Mode Flow chart 8](#_Toc167801869)

[Figure 4.2.2.2: Payee Mode Flow Chart 9](#_Toc167801870)

[Figure 5.1.1.1: ESP32 10](#_Toc167801871)

[Figure 5.1.2.1: R307 Fingerprint sensor module 11](#_Toc167801872)

[Figure 5.1.2.2: Working principle of fingerprint sensor 11](#_Toc167801873)

[Figure 5.1.3.1: 1.3" OLED Display 12](#_Toc167801874)

[Figure 5.1.4.1: 4x4 Matrix Keypad 12](#_Toc167801875)

[Figure 5.1.5.1: Power Supply 13](#_Toc167801876)

[Figure 5.1.6.1: 18650 li-ion battery 13](#_Toc167801877)

[Figure 5.2.1.1: Interfacing of R307 fingerprint sensor module with ESP32 14](#_Toc167801878)

[Figure 5.2.2.1:Interfacing of OLED display with ESP32 14](#_Toc167801879)

[Figure 5.2.3.1: Interfacing of 4x4 Matrix Keypad with ESP32 14](#_Toc167801880)

# CHAPTER 1

# INTRODUCTION

The current landscape of daily payments is dominated by cash and IC cards, each with its limitations. Cash transactions, while prevalent in offline scenarios, are cumbersome due to the need for change-giving. Similarly, IC cards, though offering convenience for everyday purchases, lack a robust authentication process.

Biometric payment emerges as a promising solution. Leveraging the unique biological characteristics of users for identification significantly enhances security in offline payments. Additionally, eliminating the need to carry cash or cards fosters greater convenience. Biometric payment effectively bridges the gap between security and ease of use.

The widespread adoption of biometric payment has yet to reach its full potential due to cost and technical hurdles. This project addresses these challenges by proposing a low-cost embedded biometric payment system, making this secure and convenient payment method more accessible.

The Biometric Based Payment System, detailed in this paper, offers a secure and streamlined payment experience. It utilizes biometric authentication for user verification, employing a fingerprint sensor (like the R307) for user identification. Users interact with a versatile 4x4 matrix keypad for functionalities such as payment amount entry, account balance inquiries, and fund deposits. All actions are confirmed on a clear I2C OLED display, providing real-time visual feedback.

The ESP32 microcontroller serves as the heart of the system. This powerful unit safeguards sensitive information through robust SHA-256 hashing for on-board data encryption. This encryption protects transaction details during transmission to a centralized MySQL database hosted on an Apache server. The system efficiently communicates between the ESP32 microcontroller and the server using the HTTP protocol. For user convenience, a dedicated web interface allows users to manage their payment history.

# CHAPTER 2

# LITERATURE SURVEY

**Paper 1. Bin RONG and Chang-hua LIU “A Fingerprint Payment System Based on FPGA.”**

2022 International Conference on Information System and Artificial Intelligence.

The low-cost embedded fingerprint payment solution is a practical approach to expanding the adoption of this technology. The FPGA-based system, with its keyboard for account management, VGA display for transaction information, fingerprint module for authentication, and data processing module for balance updates, presents a well-integrated and cost-effective solution. The SPI interface for communication with the database and upper management system further enhances its scalability and integration capabilities. The proposed fingerprint payment system utilizes a control centre module to orchestrate the operations of the display module, PS/2 keyboard module, and fingerprint identification module. The control centre, guided by a finite state machine, dynamically adjusts its state based on the state of the other modules. This intricate interplay between the modules ensures a seamless and secure payment process.

**Paper 2. Dileep Kumar and Yeonseung Ryu “A Brief Introduction of Biometrics and Fingerprint Payment Technology”.**

2020 Second International Conference on Future Generation Communication and Networking Symposia.

A global survey by Unisys reveals that nearly 70% of consumers worldwide support biometrics technology, such as fingerprints and voice recognition, for identity verification, especially when administered by trusted organizations like banks or healthcare providers. The preference for biometrics is underscored by 66% favouring it as the ideal method to combat fraud and identity theft. San Francisco-based company Pay by Touch has successfully enrolled over 2 million people for fingerprint-based transactions in stores, indicating a growing acceptance of biometric authentication in retail settings. Biometrics' application extends beyond commercial transactions, with schools, companies, and driver's license processes adopting fingerprint scanning for security purposes. Indivos, based in Oakland, promotes fingerprint scanners for electronic payments, emphasizing consumer convenience and cost reduction for retailers. Furthermore, Citibank's introduction of a fingerprint authentication payment service in Singapore allows credit card customers to make purchases with a simple touch of their finger at selected merchant locations, marking a notable advancement in biometric payment services and show casing the broader integration of biometrics into financial transaction.

**Paper 3. Dr. Chitra, Kiran.N, Suchira Suresh , Akarsh S.M , BhuvanTeja and Jerrin Y omas “A Biometric based Payment System by using Payee and Payer Module”.**

2019 2nd IEEE International Conference on Recent Trends in Electronics Information & Communication Technology (RTEICT).

In this system, biometric data, such as fingerprints or facial recognition, is used to authenticate the user's identity for secure transactions. The payee module is responsible for capturing and verifying the biometric data of the customer, while the payer module is used by the customer to authorize the payment using their biometric information. This technology provides an extra layer of security and convenience in making payment.

**Paper 4. R. Logeshwari, M. Saikotesw Ara, M.V. Sudeerkumar and J.S.R.K Suryateja “FINGERPRINT BASED ONLINE PAYMENT SYSTEM”.**

International Research Journal of Engineering and Technology (IRJET)., Oct 2019.

The paper advocates for heightened personal identification and proposes the use of fingerprints as a secure password and PIN for online cashless payments, particularly focusing on the Unique Identification Authority of India (UIDAI) database. By integrating fingerprint biometrics from UIDAI, which is linked to national banks, the paper suggests enhancing security in online transactions. The UIDAI is developing an Aadhaar based online confirmation system accessible to banks and service providers, aiming to reduce fraudulent access and ensure the accurate disbursement of payments, including government subsidies and pensions. Aadhaar cards, mandatory in India's private and government sectors, store reliable biometric information, making fingerprints a viable security measure. The proposed model suggests using UID's stored fingerprints as a password, leveraging its connection to banks to minimize hacking risks. The practicality of implementing fingerprints in shopping bill payments is facilitated by Aadhaar's widespread use for identity registration and linkage to multiple bank accounts.

**Paper 5. Yan Sui, Xukai Zou and Yinozi Du “Biometrics-Based Authentication: A New Approach”.**

20th International Conference on Computer Communications and Networks (ICCCN)., 2018.

Authentication is a fundamental issue to any trust-oriented computing system and a critical part in many security protocols. Performing authentication is notoriously difficult. Biometrics has been widely used and adopted as a promising authentication method due to its advantages over some existing methods, particularly, its resistance to losses incurred by theft of passwords and smart cards. However, biometrics introduces its own challenges, such as being irreplaceable once compromised. Moreover, the use of biometrics introduces privacy concern. In this paper, we propose a simple yet effective biometrics-based authentication solution. The proposed approach introduces new constructs - Reference Subject and Biometric Capsule and stores the ``difference'' (called Biometric Capsule) between the user and the Reference Subject for authentication without revealing a user's original biometric information. This approach supports replaceability and protect users' privacy. Moreover, the proposed approach creates more advantages: (a) being user-friendly without any additional burden on users and possessing one-for-all power; (b) being generic enough to be applied to various biometrics (e.g., fingerprint, face, iris) or combinations of them; and (c) being adaptive in terms of security and privacy to fit different authentication models, application requirements, available resources, and trusted or non-fully-trusted environments. The experimental results on iris validate its performance and prove it a practical mechanism.

# CHAPTER 3

# PROBLEM STATEMENT

* Traditional payment systems are vulnerable to security breaches, as cards can be cloned, PINs can be stolen, and sensitive payment information can be compromised.
* Remembering multiple passwords and PINs for different cards and accounts can be inconvenient and can lead to declined transactions when forgotten.
* Physical cards can be lost or stolen, creating hassle and potential financial losses if they fall into the wrong hands.

# CHAPTER 4

# OBJECTIVES AND METHODOLOGY

## OBJECTIVES

* To integrate and evaluate sensor modules for fingerprint capture, ensuring compatibility with the ESP32 environment.
* To develop a secure system for users to link their fingerprint data with their payment information.
* To enable secure and fast transactions using fingerprint authentication for payment processing.

## METHODOLOGY

### Block Diagram

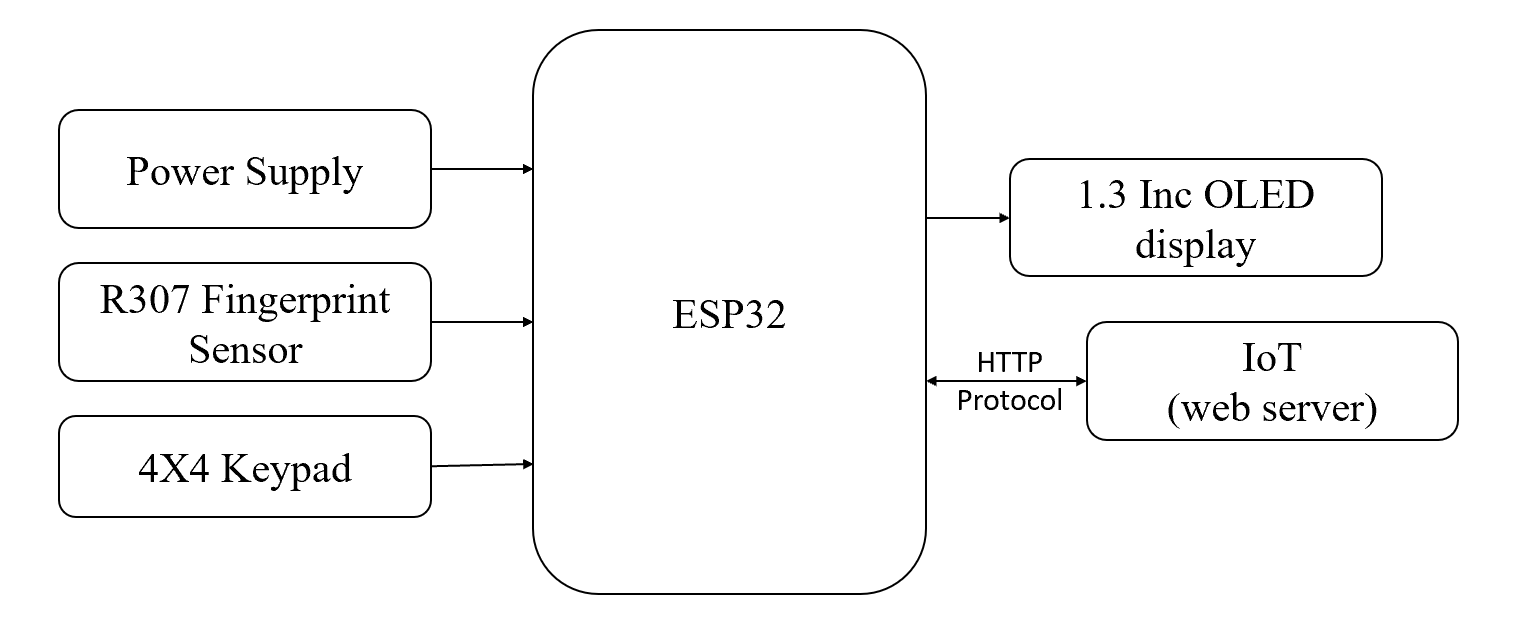


Figure 4.2.1.1: Block Diagram

* Initially the user must register their fingerprint ID.
* Once the fingerprint ID is registered the user can access their wallet for transitions.
* Basically, in payee mode fingerprint ID is being verified and if it is matched the user further processes for making transition.
* After authorisation, the user will be prompted with four options on OLED display as follows:
  + Balance
  + Add money.
  + Pay
  + Exit
* The user can enter their options from 4x4 matrix keypad.
* From the above selected options for the data will be displayed on OLED display.
* In balance option the user can view their current wallet balance and the users can add money to their wallet by choosing add money option.
* In pay option the user can transfer money from his wallet (=< Ballance). Then the transaction will be successful.

### Flow Chart

#### Enrolment Mode

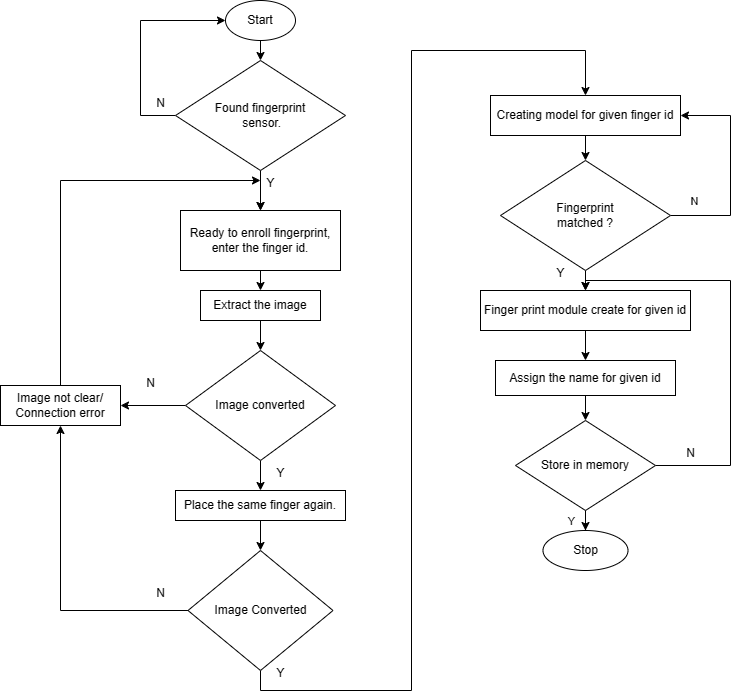


Figure 4.2.2.1: Enrolment Mode Flow chart

#### Payee Mode

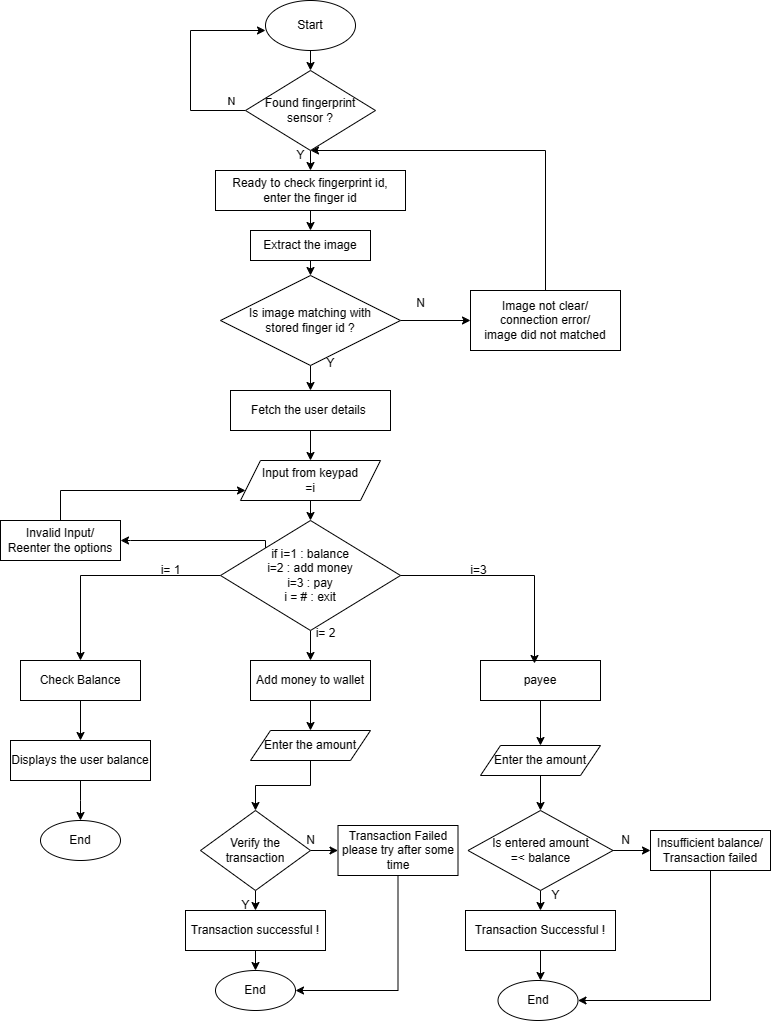


Figure 4.2.2.2: Payee Mode Flow Chart

# CHAPTER 5

# HARDWARE DESCRIPTION AND SOFTWARE DESCRIPTION

## HARDWARE REQUIREMENTS

### ESP32

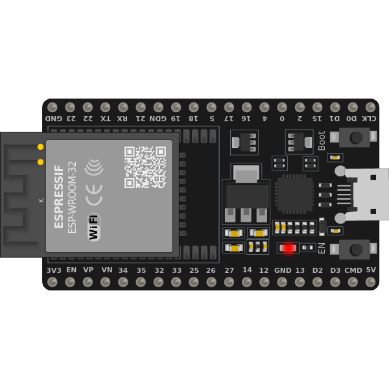


Figure 5.1.1.1: ESP32

* The ESP32 is a series of low-cost, low-power System-on-Chip (SoC) microcontrollers with integrated Wi-Fi and Bluetooth capabilities.
* Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
* 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
* 34 Programmable GPIOs.
* Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
* 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
* Secure Boot and Flash Encryption.
* Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.
* It is used as control unit.

### R307 Fingerprint Sensor Module

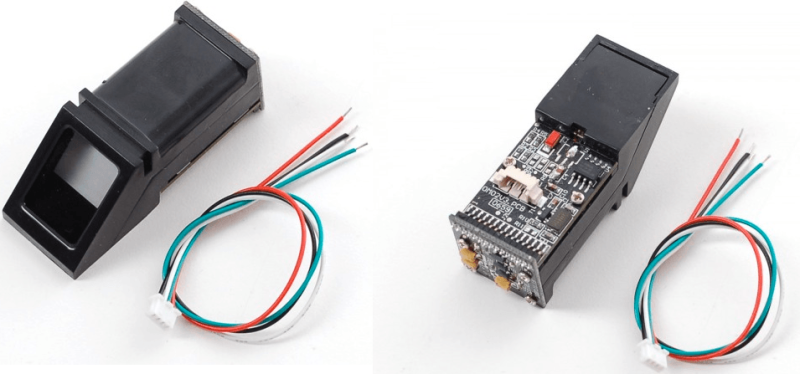


Figure 5.1.2.1: R307 Fingerprint sensor module

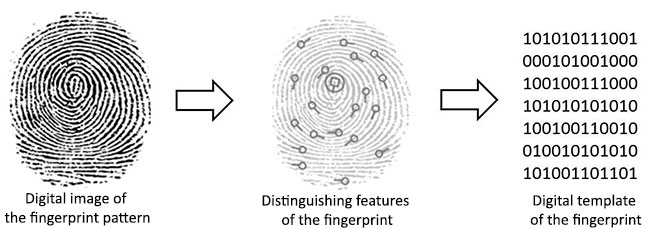


Figure 5.1.2.2: Working principle of fingerprint sensor

* The R307 fingerprint sensor module captures and recognizes fingerprints. It uses an optical sensor to capture fingerprints and has a powerful image-processing algorithm that can quickly and accurately recognize fingerprints.
* The sensor works by scanning a person’s fingerprint and creating a digital image of unique patterns of the fingertip. This image is then processed and converted into a mathematical algorithm that is stored on the sensor.
* When a fingerprint is scanned again, the sensor compares the new image to the previously stored algorithm to confirm if there is a match. If the print and algorithm are both the same, the sensor will send a signal indicating that the fingerprint is valid.
* It is used for scanning fingerprints of user.

### 1.3" OLED (Organic Light Emitting Diode) display



Figure 5.1.3.1: 1.3" OLED Display

* The 1.3″ OLED display are perfect when you need an ultra-small display. It is an OLED monochrome 128×64 dot matrix display module with I2C Interface.
* The characteristics of this display module are high brightness, self-emission, high contrast ratio, slim outline, wide viewing angle, wide temperature range, and low power consumption.
* It is used for displaying appropriate information of user.

### 4×4 matrix keypad

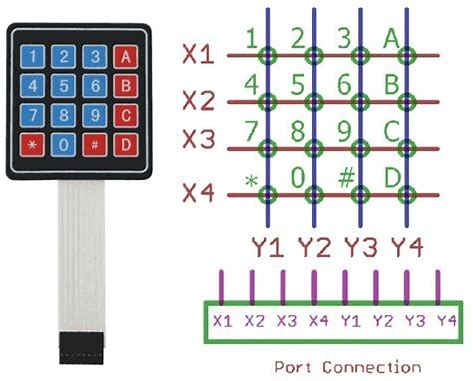


Figure 5.1.4.1: 4x4 Matrix Keypad

* The 4×4 matrix keypad is an input device, it usually used to provide input value in a project.
* It has 16 keys in total, which means it can provide 16 input values.
* It used only 8 GPIO pins of a microcontroller.
* It is used to take input from the user.

### Power Supply



Figure 5.1.5.1: Power Supply

* 5 Volt 2 Amp Power Adapter takes an AC INPUT of 100-240V and gives 5V 2A DC output.

### 18650 li-ion battery



Figure 5.1.6.1: 18650 li-ion battery

* The 18650 battery has a standardized cylindrical shape with dimensions of 18mm in diameter and 65mm in length. Its capacity typically ranges from 1800mAh to 3500mAh, with the most common being around 2600mAh to 3000mAh.
* The nominal voltage of an 18650 battery is 3.7 volts, and it is fully charged at 4.2 volts. The discharge cutoff voltage is usually around 2.5 to 3.0 volts.

## HARDWARE INTERFACE

### R307 fingerprint sensor module interface with ESP32

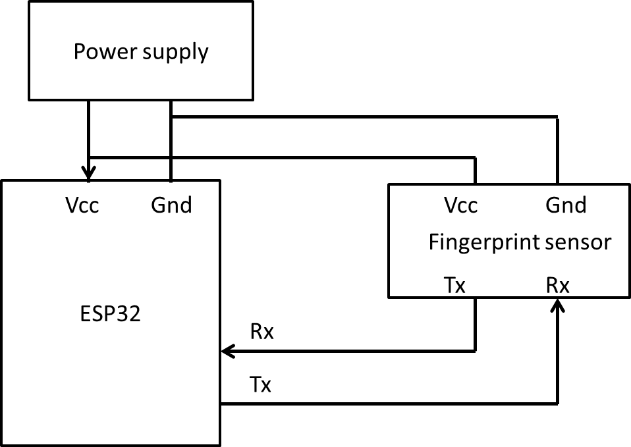


Figure 5.2.1.1: Interfacing of R307 fingerprint sensor module with ESP32

### I2C OLED display interface with ESP32

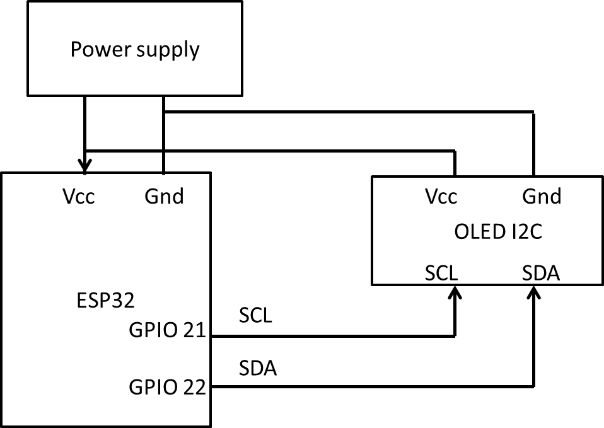


Figure 5.2.2.1:Interfacing of OLED display with ESP32

### 4×4 matrix keypad interface with ESP32

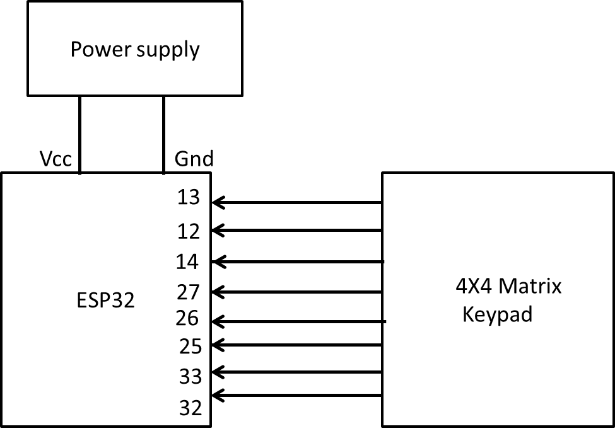


Figure 5.2.3.1: Interfacing of 4x4 Matrix Keypad with ESP32

## SCHEMATIC DIAGRAM

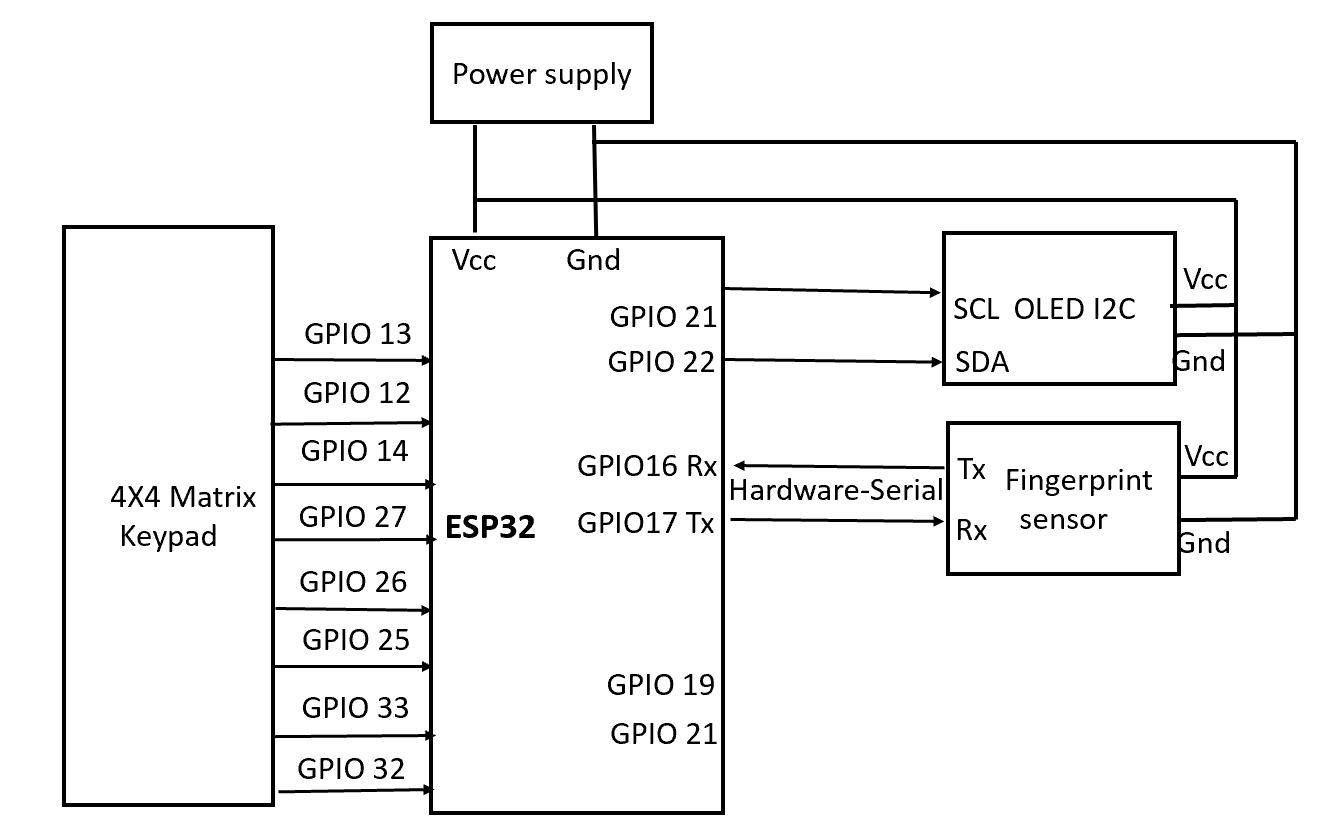


Figure5.3.1: Schematic diagram

## SOFTWARE REQUIREMENTS

### Arduino IDE

Arduino IDE is an open-source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing, and compiling the code in the environment. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a micro-controller on the board that is programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

### Hypertext Transfer Protocol (HTTP)

Is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response. HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests.

HTTP is an extensible protocol that relies on concepts like resources and Uniform Resource Identifiers (URIs), simple message structure, and client-server communication flow. On top of these basic concepts, numerous extensions have been developed over the years that add updated functionality and semantics with new HTTP methods or headers.

### Apache

Is a popular open-source, cross-platform web server that is, by the numbers, the most popular web server in existence. It is actively maintained by the Apache Software Foundation. It is fast, reliable, and secure, and runs on 31% of web servers, while an alternative, NGINX, runs on 34%. Apache powers the behind-the-scenes aspects of serving website files to visitors. It has modules which add more functions to its software, such as MPM (for handling multi-processing modes) or mod\_ssl for enabling SSL v3 and TLS support ¹. Some common features seen in Apache include: .htaccess, IPv6, FTP, HTTP/2, Perl, Lua, and PHP, bandwidth throttling, WebDAV, load balancing, URL rewriting, session tracking, and geolocation based on IP address.

### SQL (Structured Query Language)

SQL (Structured Query Language) is a domain-specific language used for managing data held in a relational database management system (RDBMS). It is a standard language for storing, manipulating, and retrieving data in databases. SQL is used to create, modify, and query databases. It is widely used in the industry and is supported by many database management systems such as MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and more.

SQL has many uses and is used in various industries such as marketing, healthcare, finance, and data science. It is used for data and business analytics, development, and data science. For instance, marketing teams often target customers and release promotions based on user data collected by the organization. Often, this data is stored in large databases and must be queried before marketers can use it.

### PHP (Hypertext Preprocessor)

PHP (Hypertext Preprocessor) is a widely used, open-source scripting language that is especially suited for web development. It was created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. PHP is a server-side language that can be embedded into HTML. It is used to create dynamic web pages, web applications, and web services. PHP is compatible with almost all servers used today, including Apache, IIS, and more. It is also supported by many database management systems such as MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and more.

PHP is a popular language in the industry and is used in various industries such as marketing, healthcare, finance, and data science. It is used for data and business analytics, development, and data science. For instance, marketing teams often target customers and release promotions based on user data collected by the organization. Often, this data is stored in large databases and must be queried before marketers can use it.

### HTML (Hypertext Markup Language)

HTML (Hypertext Markup Language) is a markup language used to create web pages. It is the standard language used to create web pages and is used to define the structure and content of a web page. HTML consists of a series of elements that define the different parts of a web page, such as headings, paragraphs, images, and links.

### JavaScript (JS)

JavaScript (JS) is a lightweight, interpreted programming language that is widely used for web development. It is a client-side scripting language that runs on the user's web browser. JavaScript is used to create dynamic and interactive web pages. It is also used in server-side programming with Node.js. JavaScript is compatible with almost all servers used today, including Apache, IIS, and more.

## SOFTWARE FLOW

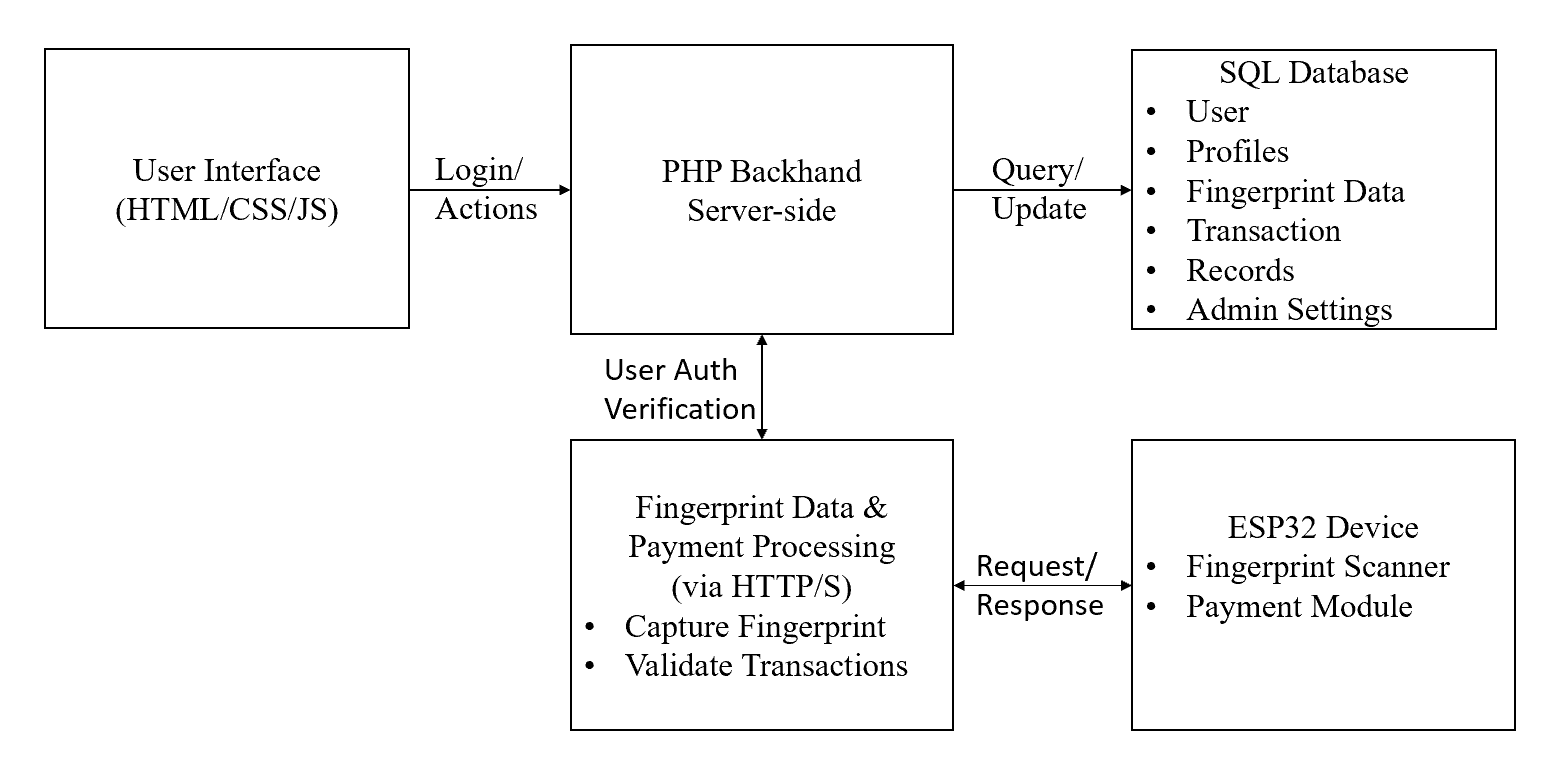
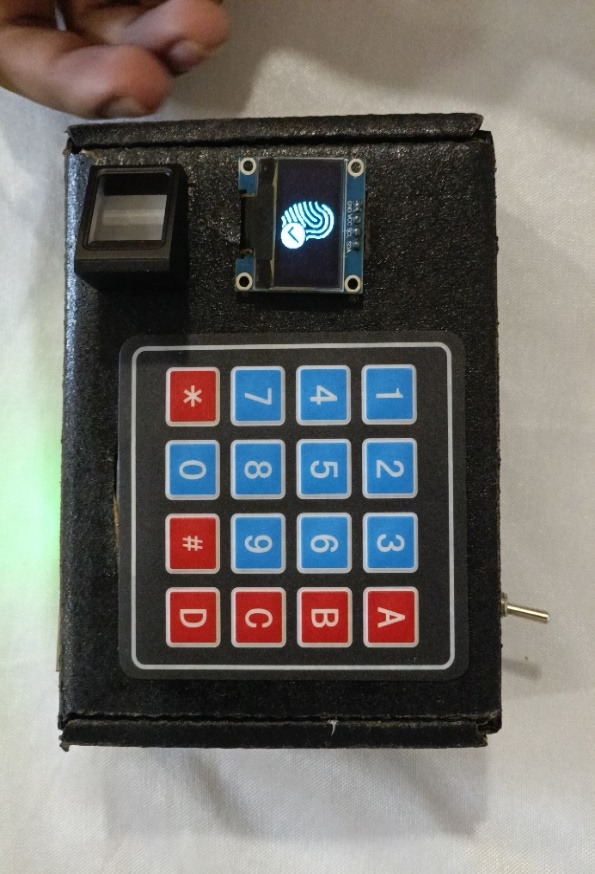
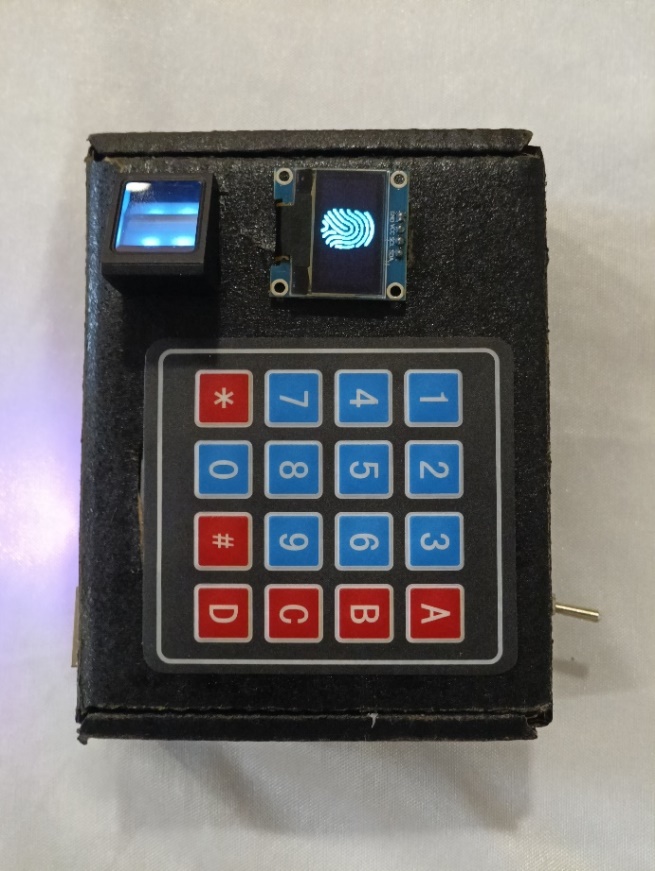
****

Figure 5.5.1: Software Flow

# CHAPTER 6

# RESULT AND DISCUSSION



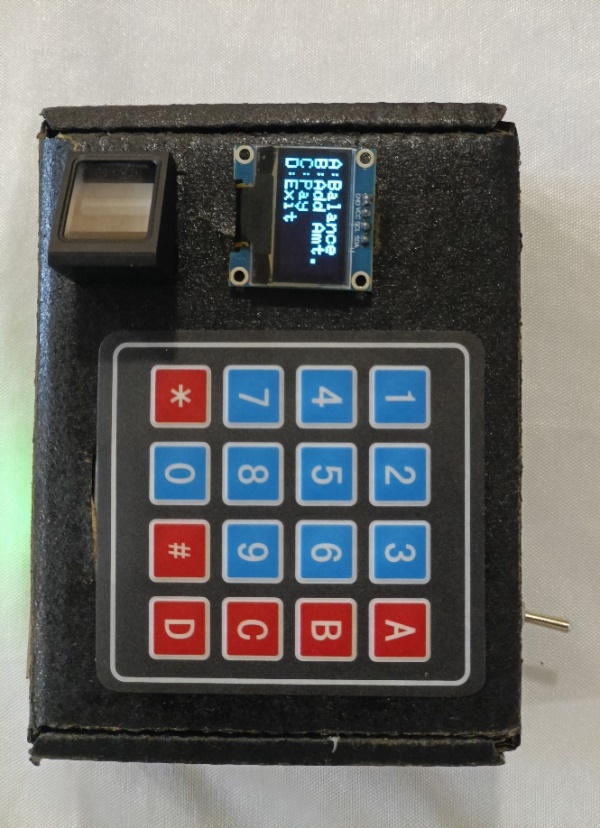


Figure6.1: Final Setup of the Project

# APPLICATIONS

**Retail Sector:**

* Point-of-Sale (POS) Systems: Enable secure and convenient transactions for customers and merchants.
* Customer Loyalty Programs: Link biometric authentication to loyalty accounts for seamless reward redemption.
* Inventory Management: Use biometric authentication for employee access to sensitive areas or to track inventory movements.

**Financial Services:**

* Banking: Authenticate customers for ATM withdrawals, online banking, and account access.
* Mobile Payments: Secure mobile wallets and payment apps with biometric authentication.
* Identity Verification: Validate customer identities during account opening and verification processes.

**Healthcare:**

* Patient Identification: Ensure accurate patient identification for accessing medical records and treatment.
* Medical Billing: Securely process payments for medical services using biometric verification.
* Pharmacies: Enable secure prescription pickups and payments with fingerprint authentication.

**Transportation:**

* Public Transit: Enable contactless fare payment and ticketing systems using biometric authentication.
* Ride-Sharing Services: Ensure driver and passenger safety by verifying identities before starting a trip.
* Airport Security: Enhance airport security with biometric-based passenger identification and boarding processes.

# ADVANTAGES

* Enhanced security comes from unique biometric identifiers that are nearly impossible to forge.
* Increased convenience is achieved by eliminating the need to carry cards, cash, or remember pins.
* Faster transactions result from swift biometric scans that streamline the checkout process.
* Reduced chance of fraud is due to the inherent security of biometric data, making it more difficult for criminals to steal.

# LIMITATIONS

* Privacy concerns exist because biometric data is highly sensitive and could be misused if security breaches occur.
* Potential for false rejections due to cuts, bruises, illness, or changes to a person's biometric identifiers over time.
* High implementation costs associated with specialized hardware and software required to implement biometric systems effectively.
* Limitations for certain users as individuals with certain physical disabilities may find it difficult to use these systems.
* Vulnerability to sophisticated spoofing where advanced techniques can be used to create fake biometric data.

# CONCLUSION

Biometric Based Payment System designed to strike a balance between security, convenience, and affordability. The system leverages fingerprint authentication, a well-established and reliable biometric modality, for robust user verification. A user-friendly interface, featuring a keypad and an OLED display, ensures intuitive interaction during the enrolment and payment processes. To safeguard sensitive user information, the system employs SHA-256 encryption throughout. This encryption protects fingerprint templates in storage and ensures data integrity during transmission to a centralized MySQL database. The system communicates efficiently with the central server using the HTTP protocol, enabling features like transaction history management through a web interface. Finally, the implementation prioritizes affordability by carefully selecting readily available components such as the ESP32 microcontroller and the R307 fingerprint sensor, making this solution accessible for wider adoption.

# FUTURE SCOPE

**Enhanced Security:**

Integration of additional biometric modalities (e.g., iris scan, facial recognition) can create a multi-factor authentication system, potentially offering even stronger security safeguards.

**Improved Offline Functionality:**

Expanding offline capabilities would broaden the system's applicability to scenarios with intermittent network connectivity. This could involve secure local storage of encrypted transaction data and robust synchronization mechanisms.

**Decentralized Security with Blockchain:**

Utilizing blockchain technology could introduce a decentralized approach to user identity and transaction management, potentially enhancing security and transparency.

**Advanced User Convenience Features:**

The system's functionality could be extended to incorporate features such as balance top-up functionalities, loyalty program integration, or integration with existing mobile payment ecosystems.

# REFERENCES

1. Bin RONG and Chang-hua LIU “A Fingerprint Payment System Based on FPGA,” 2022 International Conference on Information System and Artificial Intelligence.
2. Dileep Kumar and Yeonseung Ryu “A Brief Introduction of Biometrics and Fingerprint Payment Technology”, 2020 Second International Conference on Future Generation Communication and Networking Symposia.
3. Dr.Chitra, Kiran N, Suchira Suresh, Akarsh S.M ,BhuvanTeja and Jerrin Yomas “A Biometric based Payment System by using Payee and Payer Module”, 2019 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT).
4. R.Logeshwari, M. Saikotesw Ara, M.V. Sudeerkumar and J.S.R.K Suryateja “FINGERPRINT BASED ONLINE PAYMENT SYSTEM”, International Research Journal of Engineering and Technology (IRJET) ., Oct 2019.
5. Yan Sui, Xukai Zou and Yinozi Du “Biometrics-Based Authentication: A New Approach,” 20th International Conference on Computer Communications and Networks (ICCCN)., 2018.

# APPENDIX A- Data Sheets

**ESP32:**

|  |  |  |  |
| --- | --- | --- | --- |
| Pin Name | Pin Number | Function Description | |
| EN | 1 | Enable (active high) | |
| IO36 | 2 | GPIO36, input only, ADC1\_CH0 | |
| IO39 | 3 | GPIO39, input only, ADC1\_CH3 | |
| IO34 | 4 | GPIO34, input only, ADC1\_CH6 | |
| IO35 | 5 | GPIO35, input only, ADC1\_CH7 | |
| IO32 | 6 | GPIO32, ADC1\_CH4, RTC\_GPIO | |
| IO33 | 7 | GPIO33, ADC1\_CH5, RTC\_GPIO | |
| IO25 | 8 | GPIO25, DAC1, ADC2\_CH8 | |
| IO26 | 9 | GPIO26, DAC2, ADC2\_CH9 | |
| IO27 | 10 | GPIO27, ADC2\_CH7, RTC\_GPIO | |
| IO14 | 11 | GPIO14, ADC2\_CH6, HSPI\_CLK | |
| IO12 | 12 | GPIO12, ADC2\_CH5, HSPI\_MISO | |
| GND | 13 | Ground |  |
| IO13 | 14 | GPIO13, ADC2\_CH4, HSPI\_MOSI | |
| IO9 | 15 | GPIO9, touch 2 | |
| IO10 | 16 | GPIO10, touch 3 | |
| IO23 | 17 | GPIO23, VSPI\_MOSI | |
| IO22 | 18 | GPIO22, I2C\_SCL | |
| TXD0 | 19 | UART0\_TX |  |
| RXD0 | 20 | UART0\_RX |  |
| IO21 | 21 | GPIO21, I2C\_SDA | |
| IO19 | 22 | GPIO19, VSPI\_MISO | |
| IO18 | 23 | GPIO18, VSPI\_CLK | |
| IO5 | 24 | GPIO5, VSPI\_CS0 | |
| IO17 | 25 | GPIO17, UART2\_TX | |
| IO16 | 26 | GPIO16, UART2\_RX | |
| IO4 | 27 | GPIO4, ADC2\_CH0, HSPI\_CS0 | |
| IO0 | 28 | GPIO0, strapping pin | |
| IO2 | 29 | GPIO2, ADC2\_CH2, HSPI\_WP | |
| IO15 | 30 | GPIO15, ADC2\_CH3, HSPI\_HOLD | |
| IO8 | 31 | GPIO8, touch 0 | |
| IO7 | 32 | GPIO7, touch 1 | |
| IO6 | 33 | GPIO6, touch 4 | |
| IO3 | 34 | GPIO3, touch 5 | |
| IO1 | 35 | GPIO1, touch 6 | |
| IO45 | 36 | GPIO45, input only | |
| IO48 | 37 | GPIO48, input only | |
| IO46 | 38 | GPIO46, input only | |
| IO47 | 39 | GPIO47, input only | |
| IO43 | 40 | GPIO43, input only | |
| IO44 | 41 | GPIO44, input only | |
| GND | 42 | Ground |  |
| 3V3 | 43 | 3.3V power supply | |
| VDD | 44 | 5V power supply | |
| IO38 | 45 | GPIO38, input only | |
| IO37 | 46 | GPIO37, input only | |

**ESP32 Wi-Fi Features:**

|  |  |
| --- | --- |
| Feature | Description |
| Standard | IEEE 802.11 b/g/n |
| Frequency Range | 2.4 GHz to 2.5 GHz |
| Data Rates | 802.11b: 1, 2, 5.5, 11 Mbps<br>802.11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps<br>802.11n: up to 150 Mbps |
| Channel Bandwidth | 20 MHz, 40 MHz |
| Security | WPA/WPA2/WPA3-Enterprise/Personal, WPS |
| Modes | Station, SoftAP, SoftAP + Station |
| Power Management | Modem Sleep, Light Sleep, Deep Sleep |
| Network Protocols | TCP/IP, HTTP, HTTPS, FTP, DHCP, DNS, SNTP, Telnet, mDNS |
| Max Connections | Up to 10 stations |
| Antenna | Integrated, External (via U.FL connector, depending on the module) |
| Transmit Power | 20 dBm (max) |

**R307 Fingerprint Sensor Module:**

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Function Description |
| VCC | 1 | Power Supply (typically 3.6V to 6.0V) |
| GND | 2 | Ground |
| TXD | 3 | UART Transmit Data (to MCU RX) |
| RXD | 4 | UART Receive Data (from MCU TX) |
| TCH | 5 | Touch pin (can be used for wake-up) |
| RST | 6 | Reset pin (active low) |

**1.3” OLED display:**

|  |  |  |  |
| --- | --- | --- | --- |
| Pin Name | Pin Number | Function Description | |
| GND | 1 | Ground |  |
| VCC | 2 | Power Supply (usually 3.3V or 5V) | |
| SCL | 3 | I2C Clock (SCL) | |
| SDA | 4 | I2C Data (SDA) | |

**4x4 Keypad:**

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Function Description |
| R1 | 1 | Row 1 |
| R2 | 2 | Row 2 |
| R3 | 3 | Row 3 |
| R4 | 4 | Row 4 |
| C1 | 5 | Column 1 |
| C2 | 6 | Column 2 |
| C3 | 7 | Column 3 |
| C4 | 8 | Column 4 |

# APPENDIX B – CODE

#include <WiFi.h>

#include <AsyncTCP.h>

#include <ESPAsyncWebServer.h>

#include "mbedtls/md.h"

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <Keypad.h>

#include <Adafruit\_Fingerprint.h>

#include <HardwareSerial.h>

HardwareSerial mySerial(2); // Use UART2 for ESP32 (Pin 1=RX, 12=TX)

Adafruit\_Fingerprint finger(&mySerial);

#define OLED\_ADDRESS 0x3C

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define SDA\_PIN 17

#define SCL\_PIN 18

const byte ROWS = 4; // Four rows

const byte COLS = 4; // Four columns

char keys[ROWS][COLS] = {

{ '1', '2', '3', 'A' },

{ '4', '5', '6', 'B' },

{ '7', '8', '9', 'C' },

{ '\*', '0', '#', 'D' }

};

byte rowPins[ROWS] = { 14, 13, 12, 11 }; // Example GPIO pins

byte colPins[COLS] = { 10, 9, 46, 3 }; // Example GPIO pins

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

#define OLED\_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

const char \*ssid = "Biometric\_Based\_Payment\_System";

const char \*password = "123456789";

struct User {

int id;

String name;

float balance;

String password;

bool isAdmin;

};

User users[10];

int userCount = 0;

AsyncWebServer server(80);

String hashPassword(const char \*password) {

String salt = "123"; // Simple fixed salt for demonstration

String salting = String(password) + salt;

unsigned char hash[32];

char outputBuffer[65];

mbedtls\_md\_context\_t ctx;

mbedtls\_md\_type\_t md\_type = MBEDTLS\_MD\_SHA256;

mbedtls\_md\_init(&ctx);

mbedtls\_md\_setup(&ctx, mbedtls\_md\_info\_from\_type(md\_type), 0);

mbedtls\_md\_starts(&ctx);

mbedtls\_md\_update(&ctx, (const unsigned char \*)salting.c\_str(), salting.length());

mbedtls\_md\_finish(&ctx, hash);

mbedtls\_md\_free(&ctx);

for (int i = 0; i < 32; i++) {

sprintf(outputBuffer + (i \* 2), "%02x", hash[i]);

}

return String(outputBuffer);

}

void setup() {

Serial.begin(115200);

Wire.begin(SDA\_PIN, SCL\_PIN); // Start I2C communication

neopixelWrite(RGB\_BUILTIN, RGB\_BRIGHTNESS, 0, 0); // Red

delay(1000);

while (!Serial)

; // Wait for console to open, necessary for native USB

mySerial.begin(57600, SERIAL\_8N1, 1, 2); // Set up the sensor serial connection

finger.begin(57600); // Initialize fingerprint sensor at its operating baud rate

if (finger.verifyPassword()) {

Serial.println("Found fingerprint sensor!");

} else {

Serial.println("Did not find fingerprint sensor :(");

while (1) { delay(1); }

}

Serial.println("Waiting for valid finger...");

WiFi.softAP(ssid, password);

IPAddress IP = WiFi.softAPIP();

Serial.println("AP IP address: " + IP.toString());

users[userCount++] = { 1, "Shreyas", 100.0, hashPassword("adminpass"), true };

users[userCount++] = { 2, "Suhas", 150.5, hashPassword("pass1"), false };

users[userCount++] = { 3, "Vyshnavi", 200.75, hashPassword("pass2"), false };

users[userCount++] = { 4, "Chitra", 300.0, hashPassword("pass3"), false };

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request) {

String message = "";

if (request->hasParam("message")) {

message = request->getParam("message")->value();

}

request->send(200, "text/html", generateAdminDashboardPage(message));

});

server.on("/login", HTTP\_POST, [](AsyncWebServerRequest \*request) {

if (!request->hasParam("id", true) || !request->hasParam("password", true)) {

request->send(400, "text/html", "Bad Request - Missing Parameters!");

return;

}

int id = request->getParam("id", true)->value().toInt();

String password = request->getParam("password", true)->value();

if (authenticateUser(id, hashPassword(password.c\_str()))) {

User &user = users[id - 1]; // Adjust index for zero-based array

if (user.isAdmin) {

request->redirect("/");

} else {

request->send(200, "text/html", generateDashboardPage(user));

}

} else {

request->send(401, "text/html", "Unauthorized - Incorrect ID or Password!");

}

});

server.on("/adduser", HTTP\_POST, [](AsyncWebServerRequest \*request) {

if (userCount < 10 && request->hasParam("name", true) && request->hasParam("balance", true) && request->hasParam("password", true)) {

String name = request->getParam("name", true)->value();

float balance = request->getParam("balance", true)->value().toFloat();

String password = hashPassword(request->getParam("password", true)->value().c\_str());

users[userCount++] = { userCount + 1, name, balance, password, false };

request->redirect("/?message=User added successfully");

} else {

request->redirect("/?message=Error or maximum user limit reached");

}

});

server.on("/addmoney", HTTP\_POST, [](AsyncWebServerRequest \*request) {

if (request->hasParam("id", true) && request->hasParam("amount", true)) {

int id = request->getParam("id", true)->value().toInt();

float amount = request->getParam("amount", true)->value().toFloat();

for (int i = 0; i < userCount; i++) {

if (users[i].id == id) {

users[i].balance += amount;

break;

}

}request->redirect("/?message=Money added successfully");

} else {

request->send(400, "text/html", "Missing parameters!");

}

});

server.on("/removeuser", HTTP\_GET, [](AsyncWebServerRequest \*request) {

if (request->hasParam("id")) {

int id = request->getParam("id")->value().toInt();

removeUser(id);

request->redirect("/?message=User removed successfully");

} else {

request->send(400, "text/html", "Missing user ID for removal.");

}

});

server.on("/logout", HTTP\_POST, [](AsyncWebServerRequest \*request) {

request->send(200, "text/html", loginPage());

});

server.begin();

if (!display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDRESS)) {

Serial.println(F("SSD1306 allocation failed"));

for (;;)

; // Don't proceed, loop forever

} display.clearDisplay();

display.drawBitmap(0, 0, logo\_bmp, LOGO\_WIDTH, LOGO\_HEIGHT, 1);

display.display();

neopixelWrite(RGB\_BUILTIN, RGB\_BRIGHTNESS, 0, 0); // Red

delay(1000);

}

void loop() {

display.clearDisplay();

display.drawBitmap(0, 0, logo\_bmp, LOGO\_WIDTH, LOGO\_HEIGHT, 1);

display.display();

int userId = getUserID();

if (userId != -1) {

User &user = users[userId - 1];

displayUserName(user);

display.clearDisplay();

display.setTextSize(2);

display.setCursor(0, 0);

display.println("A:Balance");

display.println("B:Add Amt.");

display.println("C:Pay");

display.println("D:Exit");

display.display();

char key = '\0';

while (key != 'A' && key != 'B' && key != 'C' && key != 'D') {

key = keypad.getKey();

}if (key) {

switch (key) {

case 'A':

displayUserMenu(user);

break;

case 'B':

addMoney(user);

break;

case 'C':

handlePayment(user);

break;

case 'D':

break;

}

}

}delay(50);

}bool authenticateUser(int id, String hashedPassword) {

for (int i = 0; i < userCount; i++) {

if (users[i].id == id && users[i].password == hashedPassword) {

return true;

}

} return false;

}void removeUser(int id) {

int index = -1;

for (int i = 0; i < userCount; i++) {

if (users[i].id == id) {

index = i;

break;

}

} if (index != -1) {

for (int i = index; i < userCount - 1; i++) {

users[i] = users[i + 1];

}

userCount--;

}

}String loginPage() {

return R"rawliteral(

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Login</title>

<style>

body { font-family: Arial, sans-serif; background-color: #f4f4f4; display: flex; justify-content: center; align-items: center; height: 100vh; margin: 0; }

.login-container { background: white; padding: 20px 40px; border-radius: 8px; box-shadow: 0 4px 8px rgba(0,0,0,0.1); }

h2 { color: #333; }

form { margin-top: 20px; }

input[type="number"], input[type="password"], input[type="submit"] { width: 100%; padding: 10px; margin-top: 10px; border: 1px solid #ccc; border-radius: 4px; box-sizing: border-box; }

input[type="submit"] { background-color: #4CAF50; color: white; border: none; cursor: pointer; }

input[type="submit"]:hover { background-color: #45a049; }

</style>

</head>

<body>

<div class="login-container">

<h2>Login</h2>

<form action="/login" method="post">

<input type="number" name="id" placeholder="User ID" required>

<input type="password" name="password" placeholder="Password" required>

<input type="submit" value="Login">

</form>

</div>

</body>

</html>

)rawliteral";

}String generateAdminDashboardPage(String message) {

String page = R"rawliteral(

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Admin Dashboard</title>

<style>

body { font-family: Arial, sans-serif; background-color: #f4f4f4; padding: 20px; }

h1 { color: #333; }

.message { color: green; }

ul { list-style-type: none; padding: 0; }

li { margin: 10px 0; padding: 10px; background-color: #fff; border-radius: 4px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); display: flex; justify-content: space-between; align-items: center; }

form { margin-top: 20px; }

input[type="text"], input[type="number"], input[type="password"], input[type="submit"] { width: 100%; padding: 8px; margin-top: 10px; border: 1px solid #ccc; border-radius: 4px; box-sizing: border-box; }

input[type="submit"] { background-color: #4CAF50; color: white; cursor: pointer; }

input[type="submit"]:hover { background-color: #45a049; }

.button-link { background: none; border: none; color: #007BFF; text-decoration: underline; cursor: pointer; }

.button-link:hover { color: #0056b3; }

</style>

</head>

<body>

<h1>Admin Dashboard</h1>

<div class="message">)rawliteral"

+ message + R"rawliteral(</div>

<ul>

)rawliteral";

for (int i = 0; i < userCount; i++) {

page += "<li>" + String(users[i].id) + " - " + users[i].name + " - ₹" + String(users[i].balance, 2);

page += "<form style='margin: 0;' action='/removeuser' method='get'><input type='hidden' name='id' value='" + String(users[i].id) + "'><button type='submit' class='button-link'>Remove</button></form>";

page += "<form style='margin: 0;' action='/addmoney' method='post'><input type='hidden' name='id' value='" + String(users[i].id) + "'><input type='number' name='amount' placeholder='Amount' style='width: auto;'><button type='submit'>Add Money</button></form></li>";

} page += R"rawliteral(

</ul>

<h2>Add New User</h2>

<form action="/adduser" method="post">

<input type="text" name="name" placeholder="Enter user name" required>

<input type="number" name="balance" placeholder="Initial balance" required>

<input type="password" name="password" placeholder="Password" required>

<input type="submit" value="Add User">

</form>

<form action='/logout' method='post'><input type='submit' value='Logout'></form>

</body>

</html>

)rawliteral";

return page;

}String generateDashboardPage(User user) {

return R"rawliteral(

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Dashboard</title>

<style>

body { font-family: Arial, sans-serif; background-color: #f4f4f4; padding: 20px; }

h1 { color: #333; }

p { background-color: #fff; padding: 10px; border-radius: 4px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); }

form { margin-top: 20px; }

input[type="submit"] { background-color: #4CAF50; color: white; padding: 10px 20px; border: none; border-radius: 4px; cursor: pointer; }

input[type="submit"]:hover { background-color: #45a049; }

</style>

</head>

<body>

<h1>Welcome, )rawliteral"

+ user.name + R"rawliteral(</h1>

<p>Your balance is ₹)rawliteral"

+ String(user.balance, 2) + R"rawliteral(</p>

<form action='/logout' method='post'><input type='submit' value='Logout'></form>

</body>

</html>

)rawliteral";

}int getUserID() {

if (finger.getImage() == FINGERPRINT\_OK) {

if (finger.image2Tz() == FINGERPRINT\_OK) {

if (finger.fingerFastSearch() == FINGERPRINT\_OK) {

Serial.print("Found ID #");

Serial.print(finger.fingerID);

Serial.print(" with confidence of ");

Serial.println(finger.confidence);

display.clearDisplay();

display.drawBitmap(34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);

display.display();

return finger.fingerID;

} else {

display.clearDisplay();

display.drawBitmap(34, 0, FinPr\_invalid\_bits, FinPr\_invalid\_width, FinPr\_invalid\_height, WHITE);

display.display();

delay(2000);

display.clearDisplay();

display.setTextSize(2); // Normal 1:1 pixel scale

display.setTextColor(SSD1306\_WHITE); // Draw white text

display.setCursor(20, 0); // Start at top-left corner

display.println(("Invalid"));

display.setCursor(25, 20);

display.println("Finger");

display.setCursor(10, 40);

display.println("Try Again");

display.display();

Serial.println("Not Valid Finger");

delay(3000);

display.clearDisplay();

}

}

}neopixelWrite(RGB\_BUILTIN, RGB\_BRIGHTNESS, 0, RGB\_BRIGHTNESS); // purple

delay(1000);

return -1; // return -1 if no valid fingerprint found

}

void displayUserName(User &user) {

display.clearDisplay();

display.setTextSize(2); // Normal 1:1 pixel scale

display.setTextColor(SSD1306\_WHITE); // Draw white text

display.setCursor(20, 0); // Start at top-left corner

display.println("Welcome");

display.setCursor(0, 20); // Start at top-left corner

display.println(user.name);

display.display();

delay(2000);

}

void displayUserMenu(User &user) {

display.clearDisplay();

display.setTextSize(2); // Normal 1:1 pixel scale

display.setTextColor(WHITE); // Draw white text

display.setCursor(0, 0);

display.println(user.name); // Display the authenticated user's name

display.setCursor(0, 20);

display.println("Balance:");

display.setCursor(0, 40);

display.println(user.balance, 2); // Display the balance with 2 decimal places

display.display();

delay(5000);

}

void addMoney(User &user) {

float amount = 0.0;

bool amountEntered = false;

while (!amountEntered) {

amount = getKeypadNumber();

if (amount > 0.0) {

amountEntered = true;

}

}user.balance += amount;

display.clearDisplay();

display.setCursor(0, 0);

display.setTextSize(2);

display.println("Added: ");

display.println(String(amount));

display.println("Balance: ");

display.println(String(user.balance));

display.display();

delay(5000);

}

void handlePayment(User &user) {

float amount = 0.0;

bool amountEntered = false;

while (!amountEntered) {

amount = getKeypadNumber();

if (amount > 0.0) {

amountEntered = true;

}

}if (user.balance >= amount) {

user.balance -= amount;

display.clearDisplay();

display.setTextSize(2);

display.setCursor(0, 0);

display.println("Paid: ");

display.println(String(amount));

display.println("Balance: ");

display.println(String(user.balance));

neopixelWrite(RGB\_BUILTIN, 0, RGB\_BRIGHTNESS, 0); // Green

} else {

display.clearDisplay();

display.setCursor(0, 0);

display.println("INSUFF-");

display.println("-CIENT");

display.println("FUNDS");

neopixelWrite(RGB\_BUILTIN, RGB\_BRIGHTNESS, 0, 0); // Red

} display.display();

delay(5000);

}float getKeypadNumber() {

float number = 0.0;

bool isNumber = false;

display.clearDisplay();

display.setTextSize(2);

display.setCursor(0, 0);

display.println("Enter");

display.println("Amount:");

display.display();

String input = "";

while (!isNumber) {

char key = keypad.getKey();

if (key) {

if (key == '#') {

if (input.length() > 0) {

number = input.toFloat();

isNumber = true;

}

} else if (isdigit(key) || key == '.') {

input += key;

display.print(key);

display.display();

}

} delay(100);

}return number;

**COURSE OUTCOMES**

On completion of this course, we are able to:

* Get exposed to literature available for research challenges in interdisciplinary fields.
* Get an insight of real-world problems, challenges and related engineering issues to broader societal context.
* Implement their idea using modern tools & new technologies.
* Demonstrate professionalism with ethics, individual & team management and effective communication skills to comprehend & to write reports.

**CONTACT DETAILS**

|  |  |
| --- | --- |
| Name: S Suhas  USN : 4BD20EC060  Email ID : [suhassid10@gmail.com](mailto:suhassid10@gmail.com)  Mobile No: 8310556913  Address : Ballari(T),Ballari(D) |  |
| Name: Shreyas P  USN: 4BD20EC066  Email ID: [shreyasp182002@gmail.com](mailto:shreyasp182002@gmail.com)  Mobile No: 7204892099  Address: 9th cross siddaverrappa badavane, Davangere(D) |  |
| Name: Vyshnavi H K  USN: 4BD20EC090  Email ID: [vyshnavihk@gmail.com](mailto:vyshnavihk@gmail.com)  Mobile No:6361321094  Address: Mandipet Davangere(D) |  |
| Name: Chitra A  USN : 4BD21EC408  Email ID : [achithra.02@gmail.com](mailto:achithra.02@gmail.com)  Mobile No: 9380656263  Address: near durgambika temple nittuvalli Davangere(D) |  |