International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Design of Smart Electronic Voting Machine Using Arduino

Namburi Nireekshana[1], Mohammad Sohail[2], Pillalamarri Venu Gopal[3], Pegyapuram Shivakumar[4]

[1]Assistant Professor, Department of Electrical and Electronics Engineering, Methodist College of Engineering and Technology (Autonomous), Hyderabad

[2,3,4]BE III-year, Department of Electrical and Electronics Engineering, Methodist College of Engineering and Technology (Autonomous), Hyderabad

Abstract

Electronic voting systems have become an important component of modern democratic processes due to their speed, efficiency, and accuracy. However, conventional electronic voting machines are vulnerable to unauthorized access, duplicate voting, and identity fraud. This paper presents a Smart Electronic Voting Machine (SEVM) that integrates Face Recognition and QR Code Authentication with an Arduino UNO microcontroller to enhance election security and transparency. The proposed system verifies voter identity through dual authentication mechanisms before granting voting access. A Python-based Graphical User Interface (GUI) processes facial recognition and QR code validation, while the Arduino controls the voting hardware. Votes are securely recorded and stored in an encrypted database, preventing duplicate voting attempts. Experimental results demonstrate reliable voter authentication, accurate vote counting, real-time status monitoring, and automatic winner declaration. The proposed system provides a secure, low-cost, and efficient solution suitable for educational institutions, organizations, and small-scale elections.

Keywords: Electronic Voting Machine, Face Recognition, QR Code Authentication, Arduino UNO, Smart Voting System, Biometric Security

1. Introduction

In the modern digital era, democratic institutions and organizational elections require voting systems that are secure, transparent, reliable, and efficient. Elections play a crucial role in decision-making processes within governments, educational institutions, corporations, and various organizations. Traditional paper-based voting methods, although widely used, often suffer from several limitations such as lengthy vote counting procedures, human errors, ballot manipulation, unauthorized voting, and lack of transparency. These challenges can reduce public confidence in the electoral process and increase the possibility of disputes regarding election outcomes. With the rapid advancement of embedded systems, automation technologies, and digital authentication methods, there is a growing demand for intelligent voting solutions that can enhance security while simplifying election management.According to global studies on election security and digital governance, voter identity fraud, duplicate voting, and unauthorized access remain among the most significant concerns in electronic voting systems. In many cases, inadequate voter verification mechanisms allow impersonation and multiple voting attempts, compromising the integrity of elections. Furthermore, manual monitoring and vote recording procedures can introduce inaccuracies and delays in result generation. To address these challenges, modern electronic voting systems are increasingly adopting advanced authentication technologies and automated vote management techniques to ensure fairness, transparency, and accountability throughout the election process.The Design of Smart Electronic Voting Machine Using Arduino is an innovative approach aimed at improving election security and operational efficiency through the integration of embedded systems and intelligent authentication mechanisms. The proposed system utilizes Arduino UNO as the central controller and incorporates dual authentication through QR code verification and face recognition technology. Before casting a vote, each voter must successfully pass both authentication stages, ensuring that only authorized individuals are permitted to participate in the election. This multi-level verification process significantly reduces the possibility of identity fraud and unauthorized voting attempts.The system further enhances election reliability by maintaining voter records and preventing duplicate voting. Once a vote is cast, the information is securely stored in an encrypted database along with voter details and timestamps. A Python-based Graphical User Interface (GUI) manages authentication, vote recording, result processing, and system monitoring, while a 16×2 LCD display, LEDs, and buzzer provide real-time feedback and voting instructions. The integration of hardware and software components creates a comprehensive platform capable of conducting elections accurately and efficiently.The significance of this project extends beyond simple vote collection. By combining biometric verification, QR-based authentication, secure database management, and automated result generation, the system demonstrates how modern technology can strengthen democratic processes and institutional elections. Moreover, the use of cost-effective hardware components such as Arduino UNO and open-source software tools makes the system affordable and practical for educational institutions, organizations, and small-scale election environments. As the demand for secure digital voting solutions continues to grow, smart electronic voting systems have the potential to play a vital role in ensuring transparent, trustworthy, and technology-driven election management in the future[1].

1.1 Evolution of Electronic Voting Systems and the Need for Secure Elections

Elections are the foundation of democratic governance and organizational decision-making. They provide individuals with the opportunity to select representatives and participate in important decision-making processes. Traditionally, elections were conducted using paper ballots, which required significant manpower and time for vote collection and counting. Although paper-based voting systems were widely accepted, they were often associated with challenges such as ballot tampering, counting errors, invalid votes, and lengthy result declaration procedures. To overcome these limitations, Electronic Voting Machines (EVMs) were introduced to improve efficiency and accuracy in the electoral process. Over time, EVMs became increasingly popular due to their ability to reduce manual effort and accelerate vote counting. However, despite these advantages, many electronic voting systems still face security concerns related to voter authentication and unauthorized voting. The rapid advancement of embedded systems, artificial intelligence, and computer vision technologies has created opportunities for developing smarter voting solutions. Modern elections require systems that not only record votes accurately but also ensure that only eligible voters participate in the election process. Therefore, there is a growing demand for secure, intelligent, and transparent electronic voting systems capable of preventing fraud while maintaining voter confidence and election integrity[2].

1.2 Limitations of Traditional Voting Methods and Existing EVMs

Despite significant advancements in election technology, traditional voting systems and many existing electronic voting machines continue to face several limitations. Manual voting methods are often time-consuming and require extensive human supervision throughout the election process. The verification of voter identities is usually performed manually, increasing the possibility of errors and unauthorized participation. In some cases, individuals may attempt to cast multiple votes or impersonate legitimate voters, compromising election fairness. Existing EVMs have improved vote-counting efficiency but often lack advanced authentication mechanisms capable of verifying voter identity accurately. Furthermore, maintaining voter records manually can lead to inconsistencies and administrative challenges. Election officials are also required to invest considerable effort in monitoring voter activities and preventing fraudulent behavior. These limitations reduce public trust in election outcomes and create opportunities for manipulation. As election security becomes increasingly important, there is a need for intelligent voting platforms that integrate multiple verification techniques and automated monitoring systems. Such systems can significantly improve transparency, reduce human intervention, and ensure that every vote is cast and counted legitimately[3].

1.3 Towards a Smart Arduino-Based Electronic Voting Machine

The Smart Electronic Voting Machine Using Arduino is designed to address the shortcomings of traditional voting systems by incorporating advanced authentication and automation technologies. The proposed system combines Arduino UNO, QR code verification, face recognition, and a Python-based graphical user interface to create a secure and efficient election management platform. Before a voter is allowed to cast a vote, the system verifies the individual's identity through QR code scanning and facial recognition. This dual authentication process significantly reduces the possibility of unauthorized voting and identity fraud. The Arduino UNO acts as the central controller, coordinating communication between hardware components and the software platform. The Python-based GUI manages voter verification, vote recording, database operations, and result generation. Once authentication is successful, voters can cast their votes using dedicated candidate selection buttons. The system records voting information securely while maintaining complete transparency throughout the election process. By integrating modern technologies into a cost-effective platform, the Smart Electronic Voting Machine offers an innovative solution suitable for educational institutions, organizations, and small-scale elections[4].

1.4 System Architecture and Dual Authentication Methodology

The architecture of the Smart Electronic Voting Machine is designed to ensure security, reliability, and ease of operation. The system consists of Arduino UNO, a webcam for facial recognition, a QR code authentication module, push buttons for vote selection, a 16×2 LCD display, LEDs, a buzzer, and a personal computer running a Python-based GUI. The voting process begins when a voter presents a valid QR code for authentication. The system extracts and verifies the information contained in the QR code against stored voter records. After successful QR verification, the webcam captures the voter’s facial image, which is processed using computer vision algorithms to verify identity. Only when both authentication stages are completed successfully does the system enable voting access. The voter then selects a candidate using the designated push button. The Arduino records the vote and communicates with the GUI for secure storage and monitoring. Throughout the process, the LCD display provides instructions and status updates, while LEDs and buzzers offer visual and audio feedback. This dual authentication methodology enhances election security and minimizes the possibility of fraudulent voting activities[5].

1.5 Real-World Applications and Impact on Election Management

The Smart Electronic Voting Machine has significant practical applications in various election environments. Educational institutions can use the system to conduct student council elections securely and efficiently. Corporate organizations can implement the platform for employee representative elections and internal decision-making processes. Government agencies and local communities can also benefit from the system's transparent and secure voting mechanisms. By automating voter authentication and vote recording, the system reduces administrative workload and minimizes the likelihood of human errors. Election officials can monitor voting activities in real time through the graphical user interface, ensuring better control and accountability. The secure storage of voting data with timestamps improves record management and facilitates auditing when required. Furthermore, automated vote counting and result generation eliminate delays associated with manual counting procedures. The adoption of smart voting systems can enhance public confidence in election outcomes and encourage broader participation in democratic processes. As organizations increasingly embrace digital technologies, intelligent voting platforms such as this project can play a vital role in modernizing election management practices[6].

1.6 Security Mechanisms and Future-Ready Voting Infrastructure

Security is one of the most important aspects of any electronic voting system. The proposed Smart Electronic Voting Machine incorporates multiple security mechanisms to ensure election integrity and prevent unauthorized access. The first layer of security is QR code authentication, which verifies voter identity using encoded voter information. The second layer is facial recognition, which confirms that the person presenting the QR code is the legitimate voter. Together, these authentication methods significantly reduce the risk of identity fraud and duplicate voting. The system also maintains voter records and verification status to prevent multiple voting attempts. All voting information is stored securely in an encrypted database along with timestamps and voter details. Real-time monitoring through the GUI allows election administrators to track voting activities and identify irregularities immediately. The architecture is designed to support future enhancements such as biometric fingerprint authentication, cloud-based databases, blockchain technology, and remote voting capabilities. These future-ready features ensure that the system can adapt to evolving election security requirements and technological advancements[7].

1.7 Project Objectives and Technological Innovation

The primary objective of this project is to develop a secure, transparent, and efficient electronic voting system using Arduino and modern authentication technologies. The project introduces several innovative features that distinguish it from conventional voting systems. One of the key innovations is the implementation of dual authentication through QR code scanning and facial recognition. This approach provides stronger voter verification compared to traditional methods. Another innovative aspect is the integration of Arduino hardware with a Python-based GUI, enabling seamless interaction between hardware and software components. The system also incorporates automated vote recording, encrypted database storage, and real-time monitoring capabilities. Election results are generated automatically, reducing the need for manual counting and minimizing errors. User-friendly interfaces such as LCD displays, LEDs, and buzzers improve system usability and voter experience. These technological innovations contribute to a more reliable and trustworthy election process while demonstrating the practical application of embedded systems, artificial intelligence, and computer vision technologies[8].

1.8 Problem Statement

Modern election systems face several challenges related to voter authentication, security, and transparency. Traditional voting methods often rely on manual verification procedures that are time-consuming and prone to human errors. Unauthorized voting, voter impersonation, and duplicate vote casting continue to threaten election integrity in many environments. Existing electronic voting systems improve vote-counting efficiency but frequently lack robust authentication mechanisms capable of verifying voter identity accurately. Furthermore, maintaining voter records manually can create inconsistencies and increase administrative complexity. These challenges reduce public confidence in election outcomes and create opportunities for fraudulent activities. Therefore, there is a need for an intelligent electronic voting platform capable of securely authenticating voters, preventing duplicate voting, recording votes accurately, and generating transparent election results. The Smart Electronic Voting Machine Using Arduino addresses these requirements by combining QR code verification, facial recognition, automated vote recording, and secure database management within a unified system architecture.

1.9 Significance and Structural Novelty

The significance of this project lies in its ability to enhance election security, transparency, and efficiency through the integration of multiple modern technologies. Unlike conventional voting systems that depend on a single authentication mechanism, the proposed system employs both QR code verification and face recognition to establish voter identity. This dual authentication strategy greatly improves security and minimizes the risk of unauthorized participation. The project also introduces secure vote storage using encrypted databases, ensuring data integrity and preventing tampering. The combination of Arduino UNO, computer vision technology, and a Python-based GUI creates a comprehensive voting platform that is both affordable and practical. Real-time monitoring capabilities allow administrators to track voting activities and manage elections more effectively. The structural novelty of the system lies in its seamless integration of embedded hardware, biometric authentication, database management, and automated result processing within a single platform. This innovative design demonstrates how emerging technologies can be utilized to create reliable and transparent election systems

.**1.10 Objectives of the System**

The main objective of the Smart Electronic Voting Machine is to develop a secure and intelligent election management system capable of conducting fair and transparent elections. The system aims to authenticate voters through QR code scanning and facial recognition before allowing vote casting. It seeks to prevent unauthorized access, eliminate duplicate voting attempts, and maintain accurate voter records throughout the election process. Another objective is to securely store voting information in an encrypted database along with voter details and timestamps. The project also aims to provide real-time monitoring and administrative control through a Python-based graphical user interface. Automated vote counting and result generation are included to improve efficiency and reduce manual effort. Furthermore, the system is designed to provide a user-friendly voting experience through LCD displays, LEDs, and buzzer indications. Ultimately, the project seeks to demonstrate how embedded systems, artificial intelligence, and secure database technologies can be combined to create a reliable, transparent, and future-ready electronic voting solution.

2. Proposed Methodology

The operational framework of the Smart Electronic Voting Machine Using Arduino relies on a continuous, multi-stage authentication, verification, vote recording, and result management pipeline. The hardware core consists of an Arduino UNO microcontroller working in coordination with a Python-based Graphical User Interface (GUI). The controller interfaces directly with a dual-authentication architecture comprising a QR Code Verification Module and a Face Recognition System. During normal operation, the system continuously monitors voter authentication requests and validates voter credentials against the stored database. To prevent unauthorized access and duplicate voting attempts, the system executes a dual-verification algorithm in which both authentication mechanisms must be validated successfully before voting access is granted.The QR code authentication module scans the voter identification code and extracts the encoded voter information. Simultaneously, the webcam-based facial recognition module captures the voter's facial image and compares it with the registered database records. The authentication status is represented by the following verification condition:V auth = (QRvalid = TRUE) ∧ (FaceMatch = TRUE) (1) A voting enable signal (Venable) is generated only when the voter is successfully authenticated and has not previously participated in the election: Venable = Vauth ∧ (Vote Status = Not Voted) (2)When both verification conditions are satisfied, the Arduino controller activates the voting interface and enables the candidate selection buttons. The voter can then cast a vote by pressing the designated candidate button. The system immediately records the vote, updates the voter status database, and stores the transaction details along with the voter ID and timestamp in an encrypted Excel database. Simultaneously, the LCD display provides voting instructions and confirmation messages, while LEDs and buzzers generate visual and audio feedback indicating successful voter registration.

2.1 Secure Vote Recording and Election Management

Concurrent with voter registration, the Python-based GUI executes a secure vote management routine. The software updates the candidate vote count, stores voter activity logs, and prevents any further voting attempts from the same authenticated individual. This automated monitoring mechanism eliminates manual intervention and significantly improves election transparency and accuracy. By maintaining synchronized records between the Arduino controller and database system, the platform ensures reliable election management and secure vote storage.

2.1.1 Educational Institution Election Infrastructure

The system can be deployed within educational institutions for conducting student council elections, department representative selections, and various organizational voting activities. The dual-authentication framework ensures that only eligible students can participate in the election process while preventing duplicate voting attempts. The automated result generation feature further simplifies election administration and reduces counting errors.

2.1.2 Organizational and Corporate Voting Systems

For corporate organizations and private institutions, the Smart Electronic Voting Machine serves as a secure platform for conducting employee representative elections, committee selections, and policy voting procedures. Election administrators receive real-time voting information through the graphical user interface, enabling transparent monitoring and efficient management of voting operations.

2.1.3 Intelligent Public Election Management

The proposed architecture can also be adapted for community-level and local governance elections where secure voter authentication is essential. The integration of QR verification, facial recognition, encrypted vote storage, and automated result processing provides a reliable framework for modern digital election systems. The system enhances public confidence in election outcomes by ensuring transparency, security, and accuracy throughout the voting process.

2.2 System Analysis and Architectural Design

2.2.1 Architectural Framework Analysis

The Smart Electronic Voting Machine employs a structured authentication and vote management architecture. The system continuously receives authentication data from the QR code verification module and face recognition subsystem. These input blocks provide voter identification information to the processing unit, which performs database verification and voting eligibility checks. The output modules manage vote registration, result generation, LCD status display, LED indicators, and buzzer notifications. The architecture ensures secure communication between hardware and software components while maintaining accurate voter records and election statistics.

2.2.2 Peripheral Pins and Device Mapping

The Arduino UNO operates at a regulated 5V supply and serves as the primary control unit for the voting system. The candidate selection buttons are connected to dedicated digital input pins configured using internal pull-up resistors. The 16×2 LCD display communicates through the I2C interface utilizing SDA and SCL communication lines. LEDs and buzzer modules are interfaced through digital output pins for status indication and user notifications. The webcam and QR authentication modules operate through the personal computer and communicate with the Arduino controller via serial communication protocols. The Python GUI manages authentication processes, database operations, vote storage, and result generation while maintaining synchronization with the microcontroller.

2.2.3 Synthesis and Architectural Justification

The structural analysis of conventional voting systems reveals significant security vulnerabilities associated with manual verification procedures and single-factor authentication mechanisms. Existing systems often fail to prevent voter impersonation and duplicate voting effectively. The proposed architecture directly addresses these limitations through a dual-authentication framework combining QR code verification and facial recognition technology. By integrating Arduino-based hardware control, encrypted database management, real-time monitoring, and automated result processing, the system provides enhanced security, transparency, and operational efficiency. This architecture significantly reduces election fraud risks while ensuring accurate vote recording and reliable election management.

2.3 System Design

The Smart Electronic Voting Machine consists of two major functional units: the Authentication Unit and the Voting Management Unit. The Authentication Unit performs QR code verification and face recognition to validate voter identity. The Voting Management Unit comprises the Arduino controller, candidate selection buttons, LCD display, LEDs, buzzer, encrypted database, and Python-based GUI. When a voter arrives at the voting station, the Authentication Unit verifies identity credentials through dual verification. Upon successful authentication, the Voting Management Unit enables vote casting and records the selected vote securely. The system automatically updates voter records, prevents duplicate voting, and generates election results in real time. This integrated design enhances election security, minimizes human intervention, improves transparency, and ensures efficient management of modern electronic voting operations.

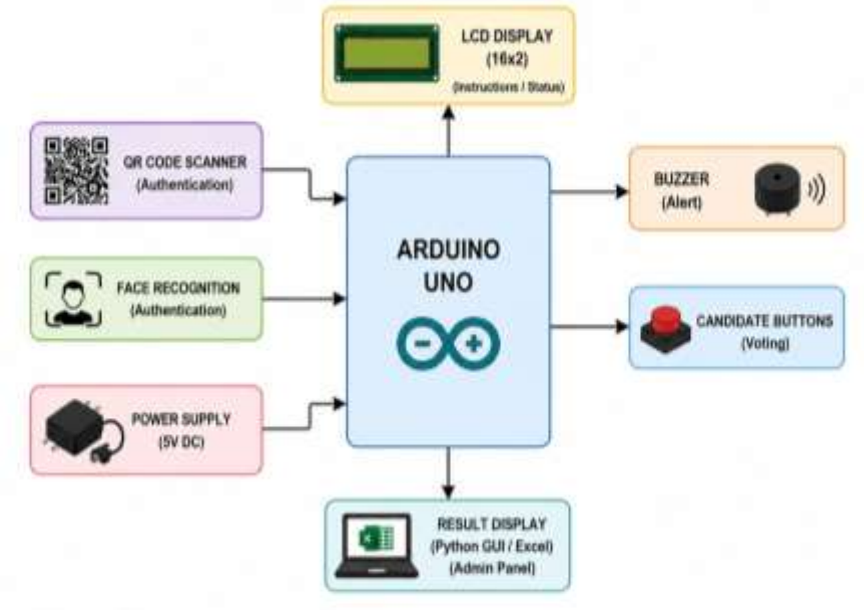

Figure 1: Block Diagram of the System

2.3.1 Voting Unit

Figure 2 Voting unit
Figure 2.: Voting Unit Side View

Figure 3 Side view of Voting Unit

2.3.2 Components List

1. Arduino UNO
2. QR Code Authentication System
3. Webcam (Face Recognition Module)
4. 16×2 LCD Display with I2C Module
5. Push Button Switches
6. Personal Computer / Laptop (Python GUI)

The Smart Electronic Voting Machine consists of six major components that work together to provide secure and transparent election management. The Arduino UNO serves as the central control unit responsible for processing authentication results and controlling the voting operations. The QR Code Authentication System acts as the first level of voter verification by validating voter identification details stored in the database. The Webcam-based Face Recognition Module provides biometric authentication by comparing the voter's facial features with registered records. A Personal Computer running a Python-based Graphical User Interface (GUI) manages voter verification, database operations, vote recording, and result generation. Once authentication is successfully completed, the Arduino enables the Push Button Switches, allowing voters to cast their votes for the desired candidate. The 16×2 LCD Display with I2C Module provides real-time instructions, authentication status, and vote confirmation messages to the user. Together, these components create a secure, reliable, and automated electronic voting platform capable of preventing unauthorized voting, eliminating duplicate votes, and ensuring accurate election management.

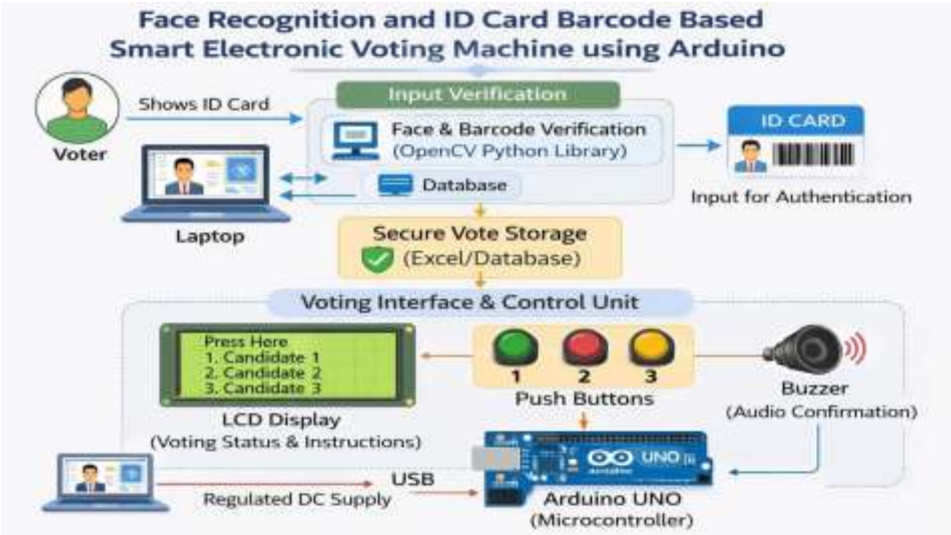

Figure 4 Block Diagram of Voting Machine

2.3.3 Arduino UNO ATMEGA328P

Figure 5 Arduino UNO ATMEGA328P

The Arduino UNO acts as the central controller of the Smart Electronic Voting Machine. It receives authentication signals from the Python-based GUI after successful QR code and face recognition verification. Once the voter is authenticated, the Arduino enables the voting buttons and records the selected vote. It also controls the LCD display, buzzer, and LED indicators to provide real-time feedback to the voter. By coordinating communication between hardware components and the software interface, the Arduino UNO ensures secure, reliable, and efficient voting operations.

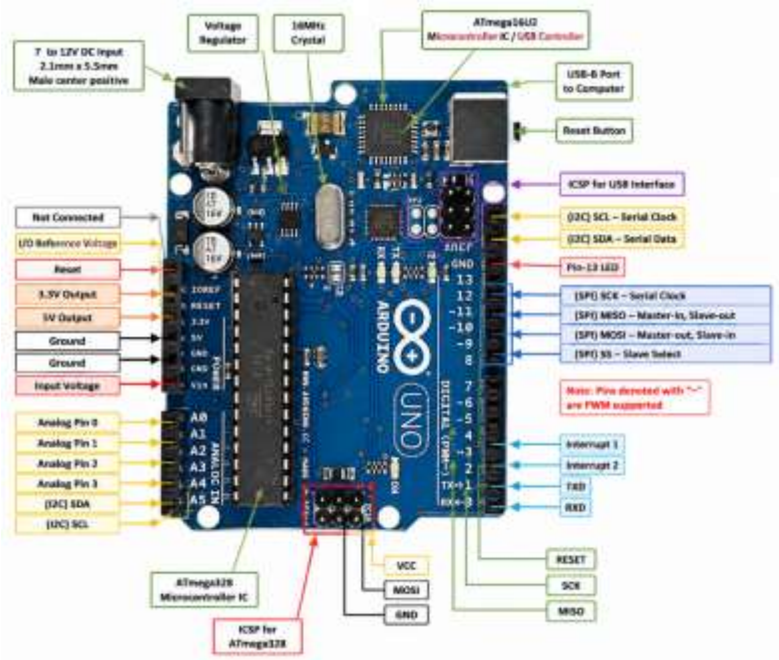

Figure 6 Arduino uno Pin Diagram

2.4 Input and Output Description of Arduino UNO (ATmega328P)

Inputs of Arduino UNO (ATmega328P):

The Arduino UNO based on the ATmega328P microcontroller receives signals and data from various external devices through its input pins. These inputs allow the controller to monitor user actions, sensor readings, and system conditions required for voting operations.

- **Digital Inputs:** Accept binary signals (HIGH or LOW) from push button switches used for candidate selection and user interaction.
- **Analog Inputs (A0–A5):** Read varying voltage levels from analog devices and sensors through the built-in Analog-to-Digital Converter (ADC).
- **Serial Communication Inputs (RX/TX):** Receive data from external devices such as a computer running the Python-based GUI through UART serial communication.
- **I2C Communication Inputs:** Utilize SDA (A4) and SCL (A5) pins to receive data from I2C-enabled devices such as the 16×2 LCD display module.
- **External Interrupt Inputs:** Detect immediate changes in input signals and respond quickly to critical events or user actions.

Outputs of Arduino UNO (ATmega328P):

The Arduino UNO sends processed information and control signals to external devices through its output pins. These outputs control system components and provide feedback to users during the voting process.

- **Digital Outputs:** Generate HIGH or LOW signals to control LEDs, buzzers, and other digital output devices.
- **LCD Display Outputs:** Send voting instructions, authentication status, vote confirmation messages, and election information to the 16×2 LCD display through the I2C interface.
- **Serial Communication Outputs (TX):** Transmit data to the Python-based GUI for voter management, vote recording, and result generation.

- **PWM Outputs:** Generate Pulse Width Modulation (PWM) signals on supported digital pins for advanced control applications if required.
- **Status Indication Outputs:** Control LEDs and buzzers to provide visual and audio feedback for successful authentication, vote registration, and system notifications.
- **Database Communication Outputs:** Transfer voting information, voter status, and election results to the connected computer system through serial communication.

The Arduino UNO acts as the central control unit of the Smart Electronic Voting Machine by receiving authentication and voting inputs, processing them using the ATmega328P microcontroller, and generating appropriate outputs to ensure secure, transparent, and reliable election management

2.5 16×2 LCD Display

Figure 7 16×2 Liquid Crystal Display (LCD)

A 16×2 Liquid Crystal Display (LCD) is used to present real-time information generated by the system. It provides a simple and user-friendly interface for displaying sensor readings, vehicle status, speed, location details, warning messages, and other operational data, enabling users to monitor the system efficiently and accurately.

2.6 I2C LCD Interface Module

Figure 8 I2C LCD Interface Module

The 16×2 LCD is interfaced with the Arduino UNO using the I2C (Inter-Integrated Circuit) protocol, enabling serial communication with minimal pin utilization. The interface requires only two lines: SDA

(Serial Data Line) and SCL (Serial Clock Line), connected to pins A4 and A5 respectively. Power is supplied through VCC (5V) and GND.

2.7 Push Button

Figure 9 Push button

Push buttons are commonly available in Normally Open (NO) and Normally Closed (NC) configurations. In the Smart Electronic Voting Machine, Normally Open push buttons are used as voting inputs. Each button is connected to a digital input pin of the Arduino UNO and configured using the internal pull-up resistor feature.When a voter successfully completes QR code and face recognition authentication, the Arduino enables the voting process. The voter can then press the desired candidate button to cast a vote. Upon detecting the button press, the Arduino records the vote, activates the corresponding indication system, and transmits the voting information to the Python-based graphical user interface.The push buttons provide a simple, reliable, and cost-effective method for vote selection. Their fast response, ease of interfacing, and durability make them suitable for electronic voting applications requiring accurate and secure user input.

2.8 Buzzer

Figure 10 Buzzer

Buzzers are broadly classified into active and passive types. An active buzzer contains an internal oscillator circuit and produces sound when a DC voltage is applied. In contrast, a passive buzzer requires an external signal from a microcontroller to generate sound.The buzzer is interfaced with Arduino through a digital output pin. When voter authentication is successful or an invalid voting attempt occurs, the controller activates the buzzer to provide immediate audio feedback. This ensures real-time user notification and enhances system usability.

3.1 Hardware Requirements

Table 1 Hardware requirements

S. No.	Hardware Component	Specification / Purpose
1	Arduino UNO	Main microcontroller used for controlling and managing the voting system
2	Webcam (Face Recognition Camera)	Captures voter facial images for biometric authentication
3	QR Code Authentication System	Scans and verifies voter QR codes for identity validation
4	16×2 LCD Display	Displays voting instructions, authentication status, and voting messages
5	I2C LCD Interface Module (PCF8574)	Enables LCD communication using only two Arduino pins (SDA and SCL)
6	Push Button Switches	Used by voters to select and cast votes for candidates
7	Buzzer	Provides audio alerts for authentication success, failure, and vote confirmation
8	LED Indicators	Provides visual indication of system status and voting operations
9	Personal Computer / Laptop	Runs the Python GUI, face recognition, QR authentication, and vote management system
10	USB Cable	Used for programming the Arduino UNO and serial communication with the computer
11	Breadboard / General Purpose PCB (Perfboard)	Used for circuit assembly and component interconnections
12	Connecting Wires and Connectors	Provides electrical connections between all hardware modules
13	5V DC Power Supply Adapter	Supplies regulated power to the Arduino and peripheral devices
14	Resistors	Used for LED protection, button interfacing, and circuit stability

3.2 Software Requirements

Table 2 software requirements

S. No.	Software	Purpose
1	Arduino IDE	Development environment used for programming and uploading code to Arduino UNO
2	Embedded C/C++	Programming language used for Arduino-based system development
3	Python	Used for developing the graphical user interface and authentication modules

S. No.	Software	Purpose
4	OpenCV Library	Performs image processing and face recognition operations
5	Face Recognition Library	Used for biometric voter authentication through facial verification
6	Tkinter GUI Framework	Provides the graphical user interface for voter management and election monitoring
7	Pandas Library	Used for voter database handling and vote record management
8	OpenPyXL Library	Stores voting records securely in Excel spreadsheets
9	PySerial Library	Enables serial communication between Arduino UNO and the Python GUI
10	Windows 10/11 or Linux OS	Platform for software development, execution, and testing
11	USB Driver for Arduino UNO	Establishes communication between the computer and Arduino board
12	Serial Monitor	Used for debugging and monitoring Arduino operations

3.3 Minimum System Requirements for Development

- **Processor:** Intel Core i3 / AMD Ryzen 3 or equivalent (Intel Core i5 recommended)
- **RAM:** 4 GB minimum (8 GB recommended for smooth face recognition processing)
- **Storage:** 1 GB free disk space minimum
- **Operating System:** Windows 10/11, Linux, or macOS
- **Webcam:** USB Camera or Integrated Webcam for face recognition

3.4 Implementation

3.4.1 Hardware Implementation

The hardware implementation of the Smart Electronic Voting Machine consists of two major units: the authentication unit and the voting management unit. The Arduino UNO microcontroller serves as the central processing unit of the system. It is responsible for controlling the voting process, receiving authentication signals, processing voter inputs, and coordinating communication between hardware and software modules. The Arduino UNO was selected due to its simplicity, reliability, and ease of interfacing with multiple peripheral devices.A 16×2 LCD display integrated with a PCF8574 I2C interface module is used to display voter instructions, authentication status, vote confirmation messages, and election-related information. The I2C module reduces the number of Arduino pins required for communication and simplifies the overall wiring structure. A breadboard or general-purpose PCB is used for mounting and interconnecting electronic components, ensuring stable electrical connections and compact circuit implementation.The authentication unit consists of a QR Code Verification System and a Webcam-based Face Recognition Module. The QR code scanner verifies voter identification details stored in the database, while the webcam captures facial images for biometric verification. These authentication mechanisms work together to ensure that only authorized voters are allowed to access the voting process. Push button switches are provided for candidate selection and vote casting. Once authentication is completed successfully, the Arduino enables the voting buttons and records the selected vote.Additional peripherals such as LEDs and a buzzer provide visual and audio feedback during authentication and voting operations. A personal computer running the Python-based Graphical User Interface (GUI) acts as the monitoring and database management unit. The complete hardware setup is integrated to provide secure voter authentication, reliable vote recording, and transparent election management.

3.4.2 Software Implementation

The software implementation of the Smart Electronic Voting Machine is designed to coordinate voter authentication, vote recording, database management, and result generation. The Arduino UNO is programmed using the Arduino IDE, which provides a user-friendly environment for developing embedded applications. The control program is written in Embedded C/C++ and uploaded to the Arduino through a USB interface. A Python-based Graphical User Interface (GUI) is developed to manage authentication and election operations.The software begins by initializing the Arduino controller, LCD display, serial communication channels, and database connections. The Python GUI then activates the QR code verification module and face recognition system. When a voter presents a QR code, the software extracts the encoded information and compares it with the registered voter database. Simultaneously, the webcam captures the voter's facial image and performs biometric matching using computer vision algorithms.After successful verification, the system enables the voting interface and allows the voter to cast a vote using the candidate selection buttons. The Arduino detects the selected candidate and transmits the voting information to the GUI. The software immediately updates the encrypted Excel database, records the vote, stores the timestamp, and changes the voter status to prevent duplicate voting attempts.

Error-handling mechanisms are incorporated to manage authentication failures, invalid QR codes, communication errors, and duplicate voting attempts. Modular programming techniques improve code readability, maintenance, and future expansion. The software architecture integrates QR authentication, face recognition, vote recording, LCD display management, and automated result generation into a unified platform. This implementation successfully demonstrates how embedded systems, computer vision, and database technologies can be combined to create a secure, transparent, and efficient electronic voting system.

3.5 Advantages

3.5.1 Enhanced Security and Authentication Reliability

A major advantage of the Smart Electronic Voting Machine is its implementation of a dual-authentication framework that combines QR code verification and facial recognition technology. Unlike traditional voting systems that rely on manual identity checks or single-level verification methods, the proposed system validates voter identity through two independent authentication stages. This significantly reduces the possibility of voter impersonation, unauthorized access, and election fraud. The integration of computer vision algorithms with QR-based identification enhances verification accuracy and ensures that only legitimate voters are permitted to participate in the election process. Furthermore, the Arduino UNO controller and Python-based GUI operate together to process authentication requests in real time, providing immediate responses and minimizing delays. The system also maintains voter records and verification status, preventing duplicate voting attempts. This combination of biometric verification, automated authentication, and secure database management results in a highly reliable voting platform capable of maintaining election integrity and improving public confidence in electoral outcomes.

3.5.2 Improved Election Efficiency and Transparency

The proposed system is specifically designed to improve the efficiency and transparency of election management. By automating voter verification, vote recording, and result generation, the system eliminates the need for extensive manual supervision and reduces the possibility of human errors. Once a voter completes the authentication process, the voting interface becomes immediately available, allowing quick and efficient vote casting. All voting records are stored securely along with voter information and timestamps, creating a transparent audit trail for election administrators. The automated result calculation feature significantly reduces the time required for vote counting and result declaration. Unlike conventional methods where counting may take several hours or even days, the proposed platform generates election results instantly upon completion of voting. The LCD display, graphical user interface, LEDs, and buzzer provide continuous feedback and operational status updates, improving user experience and system transparency. These features contribute to a faster, more accurate, and more trustworthy election process.

3.5.3 Cost-Effectiveness, Scalability, and Future Adaptability

Despite incorporating advanced authentication technologies, the Smart Electronic Voting Machine is built using affordable and widely available hardware and software components. The use of Arduino UNO, standard webcams, LCD displays, and open-source software libraries significantly reduces implementation costs compared to commercial voting platforms. The modular architecture allows individual components to be upgraded or expanded without redesigning the entire system. Educational institutions, organizations, and small-scale election environments can adopt the system with minimal infrastructure investment. Additionally, the software architecture supports future enhancements such as cloud database integration, blockchain-based vote storage, fingerprint authentication, and remote voting capabilities. The flexibility of the design ensures compatibility with evolving election requirements and technological advancements. By combining affordability, scalability, and security within a single platform, the system provides a practical solution for modern election management while encouraging the adoption of digital voting technologies across various application domains.

3.6 Applications

The key applications of the Smart Electronic Voting Machine are:

- **Secure Election Management:** The system provides secure voter authentication through QR code verification and facial recognition, ensuring that only authorized individuals can participate in elections.
- **Educational Institution Elections:** The platform can be used for conducting student council elections, departmental representative selections, and campus-wide voting activities with improved transparency and efficiency.
- **Corporate and Organizational Voting:** Companies and organizations can utilize the system for employee representative elections, committee selections, policy approvals, and decision-making processes.
- **Automated Vote Recording and Counting:** The system records votes electronically and generates results automatically, eliminating manual counting errors and reducing election processing time.
- **Prevention of Duplicate Voting:** By maintaining voter authentication records and voting status information, the system effectively prevents multiple voting attempts and election fraud.
- **Election Monitoring and Administration:** The Python-based GUI enables real-time monitoring of voter activities, vote counts, and election status, providing administrators with improved control and transparency.

- **Research and Academic Applications:** The project serves as an excellent platform for studying embedded systems, biometric authentication, computer vision, database management, and election technologies.

4. Conclusion

This project successfully addresses the growing need for secure, transparent, and efficient electronic voting systems by integrating embedded hardware, biometric authentication, and intelligent software technologies. Through the implementation of Arduino UNO, QR code verification, and facial recognition, the system effectively eliminates many of the security concerns associated with traditional voting methods. The dual-authentication mechanism ensures that only authorized voters can participate in elections, significantly reducing the risks of voter impersonation, unauthorized access, and duplicate voting. The secure storage of voting records in an encrypted database further enhances election integrity and accountability. The integration of a Python-based graphical user interface enables automated voter management, vote recording, and result generation while minimizing manual intervention and human errors. Real-time monitoring, LCD-based user guidance, and automated vote counting improve both operational efficiency and user experience. Unlike conventional voting systems that often require extensive manpower and prolonged counting procedures, the proposed platform delivers accurate election results quickly and reliably. Testing and implementation demonstrate the effectiveness of combining Arduino-based control systems with computer vision and database technologies for modern election applications. Ultimately, this project represents a significant advancement in electronic voting technology by providing a cost-effective, scalable, and future-ready solution. By enhancing election security, transparency, and reliability, the Smart Electronic Voting Machine contributes to the modernization of democratic and organizational voting processes. The system establishes a strong foundation for future developments in secure digital elections and demonstrates the practical application of emerging technologies in creating trustworthy and efficient voting infrastructures.

References

1. F. Alsuhaym, T. Al-Hadhrami, F. Saeed, and K. Awuson-David, “Toward home automation: An IoT based home automation system control and security,” in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, IEEE, 2021, pp. 1–11. Accessed: Jun. 20, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9493464/
2. P. K. Balachandran, “Smart IoT in Renewable Energy Systems,” *Recent Adv. Electr. Electron. Eng. Former. Recent Pat. Electr. Electron. Eng.*, vol. 17, no. 10, pp. 953–953, Dec. 2024, doi: 10.2174/235209651710240812115711.
3. A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, and M.-H. Yang, “IoT-Based Smart Alert System for Drowsy Driver Detection,” *Wirel. Commun. Mob. Comput.*, vol. 2021, no. 1, p. 6627217, Jan. 2021, doi: 10.1155/2021/6627217.
4. N. Nireekshana, “A POD Modulation Technique Based Transformer less HERIC Topology for PV Grid Tied-Inverter,” in *E3S Web of Conferences*, EDP Sciences, 2025, p. 01001. Accessed: Apr. 11, 2025. [Online]. Available: https://www.e3s-conferences.org/articles/e3sconf/abs/2025/16/e3sconf_icregcsd2025_01001/e3sconf_icregcsd2025_01001.html

5. N. Nireekshana, R. Ramachandran, and G. V. Narayana, “A New Soft Computing Fuzzy Logic Frequency Regulation Scheme for Two Area Hybrid Power Systems,” *Int. J. Electr. Electron. Res.*, vol. 11, no. 3, pp. 705–710, Aug. 2023, doi: 10.37391/IJEER.110310.
6. N. Nireekshana, R. Ramachandran, and G. V. Narayana, “A Novel Swarm Approach for Regulating Load Frequency in Two-Area Energy Systems,” *Int. J. Electr. Electron. Res.*, vol. 11, no. 2, pp. 371–377, Jun. 2023, doi: 10.37391/ijeer.110218.
7. S. Nižetić, P. Šolić, D. L.-I. Gonzalez-De, and L. Patrono, “Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future,” *J. Clean. Prod.*, vol. 274, p. 122877, 2020.
8. S. Pirbhulal *et al.*, “A novel secure IoT-based smart home automation system using a wireless sensor network,” *Sensors*, vol. 17, no. 1, p. 69, 2016.