

# **SMART VOTING WEB BASED APPLICATION USING FACE RECOGNITION & OTP VERIFICATION**

**A PROJECT REPORT**

*Submitted by*

**BANDI NIVAS NAIDU [RA2111003010645]**

**BODDETI MUSILI NAIDU [RA2111003010624]**

*Under the Guidance of*

**Dr. SHIJU KUMAR P. S.**

Assistant Professor, Department of Computing Technologies

*in partial fulfillment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTING TECHNOLOGIES  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  
KATTANKULATHUR - 603 203**

**MAY 2025**



Department of Computing Technologies  
SRM Institute of Science & Technology  
Own Work Declaration Form

Degree/ Course : B.Tech/ Computer Science and Engineering  
Student Name : BANDI NIVAS NAIDU, BODDETI MUSILI NAIDU  
Registration Number : RA2111003010645, RA2111003010624  
Title of Work : Smart Voting Web Based Application Using Face  
Recognition & Otp Verification

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not our own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that we have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

**DECLARATION:**

We are aware of and understand the University's policy on Academic misconduct and plagiarism and we certify that this assessment is my / our own work, except where indicated by referring, and that we have followed the good academic practices noted above.

DATE :

BANDI NIVAS (RA2111003010645)

BODDETI MUSILI (RA2111003010648)

*B. Naidu*


*B. L. Naidu*




**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**KATTANKULATHUR - 603 203**

**BONAFIDE CERTIFICATE**

Certified that 18CSP109L - Major Project report titled "Smart Voting Web Based Application Using Face Recognition and Otp Verification " is the bonafide work of "BANDI NIVAS NAIDU [RA2111003010645], BODDETI MUSILI NAIDU [RA2111003010624]" who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

  
**SIGNATURE**  
**Dr. SHIJU KUMAR P. S**  
**SUPERVISOR**  
**ASSISTANT PROFESSOR**  
Department of  
Computing Technologies

  
**SIGNATURE**  
**Dr. G. NIRANJANA**  
**PROFESSOR & HEAD**  
Department of  
Computing Technologies



  
**EXAMINER 1**

  
**EXAMINER 2**

**(Dr. K. C. Sharmu)**



## ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leonus Jesu Martin M**, Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to **Dr. M. Pushpalatha**, Professor and Associate Chairperson, School of Computing and **Dr. C. Lakshmi**, Professor and Associate Chairperson, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department **Dr. G. Niranjana**, Professor, Department of Computing Technologies, SRM Institute of Science and Technology for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators and **Dr. M. Arulprakash** Panel Head, **Dr. Snehasish Ghosh**, Assistant Professor and Panel Member, **Dr. Shiju Kumar PS**, Assistant Professor, Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. Sindhuja M**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. Shiju Kumar P. S**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his mentorship. He provided us with the freedom and support to explore the research topics of our interest. His passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

*B. N. Naidu*

**BANDI NIVAS NAIDU [RA2111003010645]**

*B. D. Naidu*

**BODDETI MUSILI NAIDU [RA2111003010624]**

## **ABSTRACT**

Online voting systems utilize digital platforms to enable voters to cast their ballots electronically via the internet. These systems are designed to increase accessibility, particularly for individuals with disabilities, expatriates, and those living in remote areas. By removing physical barriers to voting, online platforms can significantly enhance voter participation and ensure that a broader segment of the population can engage in the democratic process. In addition to improving accessibility, online voting systems offer several practical benefits. They can lead to higher voter turnout, as more people can easily access voting platforms from anywhere. Furthermore, the administrative costs associated with traditional voting methods can be reduced, and the process of counting and tabulating results can be expedited, leading to quicker announcements of election outcomes. Despite these advantages, the implementation of online voting systems comes with significant challenges. Ensuring the security of the voting process is paramount to prevent fraud and cyber-attacks. Voter privacy and system integrity must also be safeguarded through robust encryption and authentication mechanisms. Additionally, a comprehensive legal framework is necessary to address the complexities of online voting. If these challenges are effectively managed, online voting holds great potential to modernize elections and strengthen democratic participation.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>viii</b>
<b>ABBREVIATIONS</b>	<b>ix</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction to Online Smart Voting using Open CV . . . . .	1
1.2 Motivation . . . . .	2
1.3 Sustainable Development Goal of the Project . . . . .	2
<b>2 LITERATURE REVIEW</b>	<b>4</b>
2.1 Related Work . . . . .	4
2.2 Limitations Identified from Literature Survey . . . . .	5
2.3 Research Objectives . . . . .	6
2.4 Product Backlog . . . . .	7
2.5 Plan of Action . . . . .	8
<b>3 SPRINT PLANNING AND EXECUTION METHODOLOGY</b>	<b>11</b>
3.1 SPRINT I . . . . .	11
3.1.1 Objectives with user stories of Sprint I . . . . .	11
3.1.2 Functional Document . . . . .	12
3.1.3 Architecture Document . . . . .	13
3.1.4 Outcome of objectives/ Result Analysis . . . . .	14
3.1.5 Sprint Retrospective . . . . .	14
3.2 SPRINT II . . . . .	15
3.2.1 Objectives with user stories of Sprint II . . . . .	15

3.2.2	Functional Document . . . . .	16
3.2.3	Architecture Document . . . . .	17
3.2.4	Outcome of objectives/ Result Analysis . . . . .	18
3.2.5	Sprint Retrospective . . . . .	18
3.3	SPRINT III . . . . .	19
3.3.1	Objectives with User Stories of Sprint III . . . . .	19
3.3.2	Functional Document . . . . .	19
3.3.3	Architecture Document . . . . .	21
3.3.4	Outcome of objectives/ Result Analysis . . . . .	22
3.3.5	Sprint Retrospective . . . . .	22
<b>4</b>	<b>ARCHIETECTURE DESCRIPTON</b>	<b>24</b>
<b>5</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>28</b>
5.1	Project Outcomes . . . . .	28
<b>6</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>31</b>
<b>7</b>	<b>REFERENCES</b>	<b>33</b>
<b>A</b>	<b>CODING</b>	<b>36</b>
<b>B</b>	<b>CONFERENCE PUBLICATION</b>	<b>74</b>
<b>C</b>	<b>PLAGIRSIM REPORT</b>	<b>75</b>

## LIST OF FIGURES

4.1	Architecture Diagram . . . . .	25
5.1	Signup Page . . . . .	30
5.2	Architecture Diagram . . . . .	30



## **ABBREVIATIONS**

<b>AES</b>	Advanced Encryption Standard
<b>ECC</b>	Elliptic Curve Cryptography
<b>OTP</b>	One-Time Password
<b>PKI</b>	Public Key Infrastructure
<b>UI</b>	User Interface
<b>SMS</b>	Short Message Service
<b>DB</b>	Database
<b>SVM</b>	Smart Voting System
<b>ML</b>	Machine Learning (related to face recognition algorithms)
<b>ID</b>	Identification

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction to Online Smart Voting using Open CV

The implementation of e-voting in electoral processes shows a greater extent of digitization for democratization processes. Voters will now have the opportunity to place their votes over the internet using some particular platforms established for e voting to make voting easy and approachable for all stakeholders in the electoral process. The key objective of online voting systems is to make voting more accessible for people with disabilities, expatriates, and residents in areas far from the actual polling stations.

One of the key advantages of online voting systems is their potential to increase voter turnout. One of the major strengths in web-based voting systems would be that they are designed to increase the general total turnout for voting. Online voting would offer more accessible and user-friendly elections, which would suit many more people for democratic participation. Online voting would be able to reduce costs that go to the administrative procedure in such elections; its electoral process would become smooth, and election results far quicker and more accurate compared to traditional paper-based systems.

However, the acceptance of voting systems online raises immense concerns that need to be overcome to assure voters with the integrity and security of this voting process. Main issues include how not to commit fraud during elections; how to secure voters' privacy; and, by all means, defend this system against any form of cyber-attack. Encrypted robust means, authenticity mechanisms, and legal coverage are some ways in the process of curbing those risks. Despite these challenges, the potential of online voting systems to modernize elections and enhance democratic engagement is considerable, making it a promising area of development.

## **1.2 Motivation**

The issues related to traditional election voting, for which one may argue a failure and inefficiency level for accessing voting due to problems, such as data breach among others, spurred innovation regarding the Smart Voting System. As crucial the election could be through ballot process participation, there arise current electronic voting systems compromised through illegal voting impersonations unauthorized remote access and takes lengthy minutes for waiting that will not help but put opportunities ready to alter in results electronically.

Because digital technology is used in almost all aspects of life, the process of voting has to be modernized in real time to make it secure, efficient, and accessible. This project makes use of advanced technologies like facial recognition, OTP authentication, and encryption in the design of a digital voting platform to improve verification of the voter, protection against unauthorized access, and ensure integrity of votes. It is through an impeccable, secure, and user-friendly system that the Smart Voting System would serve to enhance democratic participation, prevent election fraud, and gain public trust in the electoral process.

The Smart Voting System is an initiative at modernizing voting in regard to security and efficiency from the traditional systems. Because the current methods are subject to breaches and delays, they may agitate voters and lower the turn-up. Therefore, this system has fewer possibilities of unauthorized voting; assurance of confidentiality, and secure encryptions. It maximizes election reliability while making space for all citizens.

## **1.3 Sustainable Development Goal of the Project**

This system is in consonance with United Nations Sustainable Development Goal 16: Peace, Justice, and Strong Institutions. It stresses that institutions dealing with democratic participation, among all others, must be transparent, inclusive, and accountable. The designed Smart Voting System will ensure no fraudulent activity by using biometric authentication and secure digital protocols to achieve accuracy in recorded voter identity and secured integrity of votes cast. These directly contribute to fairer and more transparent elections, strengthening public trust in the democratic process and peace and justice through reliable, fraud-resistant voting

systems.

It further supports SDG 10: Reduced Inequalities, and hence more accessible to all citizens-included are the rural dwellers or incapacitated ones. The system offers an interface that is user-friendly in very rare cases and a mobile platform that can make it easier for traditionally marginalized or underserved populations to vote. This access promotes narrowing gaps in voter involvement to ensure that a more differentiated and representative population can have voices for their opinions, thereby having decision making at all government levels be more inclusive. Finally, the project indirectly supports SDG 9: Industry, Innovation and Infrastructure: In that it shows how technological innovations might be used to make critical civil functions such as voting more effective. The Smart Voting System, through secure digital infrastructure, biometric technology, and encrypted data handling, exemplifies how sustainable innovations can modernize essential services. This focus on resilience and efficiency in electoral systems sets a precedent for using technology to improve institutional frameworks, inspiring further developments that can streamline and secure public services globally.

# CHAPTER 2

## LITERATURE REVIEW

### 2.1 Related Work

Considerable research has focused on enhancing the security and efficiency of digital voting systems through biometric authentication and encryption strategies. Ahmed et al. introduced a voting system that uses biometric authentication alongside a blockchain-based infrastructure to secure data storage and transparency. Their architecture prioritizes blockchain technology, decentralizing the voting process to improve both trust and security.

Chen et al. explored facial recognition for voter verification, demonstrating that this technology significantly minimizes the risk of identity theft or impersonation by cross-referencing voter facial characteristics with a pre-existing database.

Singh et al. proposed an approach combining biometric fingerprint recognition with OTP-based authentication to reinforce voting system security. This solution proved effective in preserving voter privacy and preventing unauthorized access, although it relied on fingerprint scanners and cellular networks, limiting its practicality in areas with limited technological infrastructure.

Li et al. developed a cryptographic voting system utilizing elliptic curve cryptography (ECC) to secure vote transmission. They argued that ECC provides high security with lower computational demands compared to other encryption methods. Despite these benefits, the system encountered usability challenges, particularly for voters unfamiliar with complex digital interfaces.

Mishra et al. investigated multi-factor authentication, integrating biometric data (facial and fingerprint recognition) with OTP verification. This dual-layer approach improved system security by mitigating biometric-only vulnerabilities, such as spoofing and database breaches. However, the study raised concerns about data privacy, specifically the protection of sensitive biometric data.

Kim et al. explored a dual-modality security approach that combines facial and voice recognition to enhance both security and usability. This method showed promising results, though voice recognition accuracy was limited in noisy environments.



Choi et al. presented a decentralized voting system that uses blockchain-based smart contracts to increase transparency and minimize vote manipulation risks. While effective, this approach faced scalability issues inherent to blockchain and the complexity of implementing smart contracts for large-scale elections.

Jones et al. conducted a comprehensive review of biometric techniques for secure voting systems, comparing fingerprint scanning, iris recognition, and facial identification. They concluded that multi-modal biometric integration offers higher security levels but at the cost of increased implementation complexity.

Yuan et al. introduced a cryptographic voting system using homomorphic encryption to protect voter confidentiality and data integrity by enabling vote processing without decryption. Although highly secure, the system's significant computational requirements posed challenges for large-scale election applications.

Finally, Patel et al. proposed an electronic voting system based on public key infrastructure (PKI) for secure voter authentication and encrypted vote transmission. Despite strengthening system security, the approach faced practical challenges with certificate management and substantial infrastructure requirements.

## **2.2 Limitations Identified from Literature Survey**

Current digital voting systems are innovative but suffer from a few key limitations. Systems that use either fingerprints or facial recognition suffer from lower accessibility and accuracy in dimly lit environments or with users who have physical impairments. Multi-modal biometric solutions may help increase reliability, but the added cost and complexity of implementation make this option impractical. The fact that blockchain-based voting has been proposed as more transparent harms its accessibility in a large-scale election. More commutative power and infrastructure resources can pose barriers to the system performance and user experience, particularly in less resourceful settings. Systems that utilize OTP-based authentication for added security also create issues, primarily in such locations with lousy mobile connectivity, which again affects access as well as reliability. Though advanced cryptographic systems, homomorphic encryption, and PKI protect data, voter interfaces become complex, ruling out voters who are not up to date with technology. Issues of privacy also arise with biometric-based voting

systems because breaches in any biometric information reveal.

While passwords may eventually be reset, these irreversible privacy risks are irreversible. The infrastructure and maintenance costs of these advanced digital voting systems may prove expensive for developing regions, which calls for a balanced approach ensuring security, accessibility, and cost-effectiveness. Another limitation found in the digital voting literature is the possible issues of data privacy. Biometric systems store sensitive personal data, such as facial scans or fingerprints, which, if compromised, would cause a major breach of privacy. Encryption techniques that have been used traditionally to secure data in transmission are still susceptible to the most advanced attacks, and whether the standards of encryption in use today are good enough for emerging threats like quantum computing is still a debate. Further, some systems implement central data storage, which even though encrypted, becomes a single point of weakness in which malicious actors can enter. The balance challenge between system usability and privacy to ensure data security is the greatest challenge; much research needs to be pursued into decentralized, secure data storage and transmission that enhances voters' identity protection and confidence in the electoral process.

## **2.3 Research Objectives**

The Smart Voting System is a next-generation digital voting platform designed to revolutionize the electoral process using modern technologies which augment security, accessibility, and efficiency. Our vision is to design an environment that is safe, easy to use, and available to every eligible voter while making sure every vote is cast, registered, collected and handled correctly and safely. The system's core goals include ensuring the identity of the voter by use of biometric face recognition and One-Time Password (OTP) verification which provides the highest level of security against access and impersonation of the user. At the same time, with its Smart Voting System who can also quickly and easily step participants in the young to elderly in the far and disabled from the elections even so the illustrative barrier is easy to use. Every vote is made in a way that prevents altering it from that moment on, making it physically impossible to manipulate the underlying system of votes and helping to prevent the ever-present suspicion of election rigging. The basic premise is that voters are offered a complete and easy way to cast their votes while electoral bodies are offered a more modern and trustable means of administra-

tion that minimizes problems experienced by voters using paper-based systems. In summary, by improving election security, efficiency, and accessibility, the Smart Voting System has set a new benchmark for digital voting solutions, calling for a more equitable and dependable electoral system.

The Smart Voting System aspires to change the electoral sphere for the better by offering a secure, transparent, and easy-to-use system for all citizens wishing to exercise their right to vote, thus making the people confident that they will be able to exercise their democratic rights. We plan on using high tech solutions such as biometric identification and encryption, while still making it easy for everyone, especially the disadvantaged, to vote.

## **2.4 Product Backlog**

The Smart Voting System's product backlog focuses on prioritizing essential features that guarantee a secure, efficient, and user-friendly voting experience. To start, user registration and authentication enable voters to sign up using personal information and ID verification through a face recognition module. This feature creates unique voter identities, which are securely stored in line with data privacy standards.

The OTP feature enhances security by sending a one-time password to voters after face verification, creating a dual-layered authentication that prevents unauthorized access and reinforces system integrity. The voting interface is intuitive and user-friendly, allowing voters to review and confirm choices with ease, ensuring accessibility across various devices. Each vote is protected through AES encryption and secure storage, supported by regular integrity checks to maintain confidentiality and prevent tampering. Robust security protocols—including strict access controls, frequent audits, and anti-spoofing measures—fortify the system against threats. Continuous usability testing and user feedback contribute to an evolving design, making the system increasingly accessible, secure, and efficient for all users.

The backlog also highlights scalability and performance optimization, preparing the system to manage high user volumes during elections while maintaining a seamless experience under heavy load. Compliance with regulations is a priority, with built-in features for transparency, such as audit logs and reporting functions, to ensure trustworthiness and adherence to election laws.

Future plans include developing a mobile application to broaden accessibility across different platforms. After each election, analysis and reporting will be conducted to assess performance and user satisfaction.

## **2.5 Plan of Action**

To implement the Smart Voting System, a structured plan of action ensures a streamlined approach, addressing key components like security, efficiency, and user convenience.

### **Phase 1: Initial Requirements and System Analysis**

In this phase, the project team will carry out a thorough analysis of the system requirements, pinpointing crucial features like user registration, biometric authentication, OTP generation, secure data storage, and a user-friendly interface. To outline the complete scope of the project, stakeholder meetings, requirements gathering sessions, and security risk assessments will be held. This analysis will also help determine the project timeline, budget, and key deliverables.

### **Phase 2: Technology and Infrastructure Selection**

The team will choose the right technologies, tools, and infrastructure components based on the requirements. They will carefully select facial recognition algorithms, OTP services, encryption protocols like AES-256, and data storage solutions to address both security and performance needs. Furthermore, they will finalize decisions about the necessary hardware, including servers and data storage solutions. This phase is essential to ensure that the system is constructed on a strong technological foundation.

### **Phase 3: System Architecture and Design**

The architectural framework of the system will be developed, incorporating modules for facial recognition, OTP verification, user authentication, a voting interface, and encrypted data storage. Each module will come with a comprehensive design document that describes its functions, data flow, and interactions. The design will feature a block diagram that illustrates the secure workflow, outlining the processes from voter registration to vote storage. Additionally, user interface designs will be completed to guarantee that the system is both accessible and user-friendly.

### **Phase 4: Development of Core Modules**

In this phase, we will start developing the key components of the system: voter authentication,

OTP verification, a user-friendly voting interface, and data encryption methods. We will create a face recognition module to guarantee accurate and reliable identification of registered voters. OTP mechanisms will be set up to offer dynamic, time-sensitive verification. The voting interface will be crafted for simplicity, making it easy for users to select candidates and confirm their votes. Additionally, we will integrate data encryption protocols to protect all voting information.

**Phase 5: Testing and Quality Assurance** Once development is over, the system will undergo large-scale testing to guarantee reliability and security. Unit tests for individual components, integration tests verifying module interoperability, and security assessments for vulnerabilities. Detailed performance tests will be performed to prove the accuracy of the facial recognition and OTP modules. The accuracy, and speed of the system will be evaluated with the accuracy will be evaluation by using a confusion matrix and the user interface will be evaluate with usability and accessibility of the user. Encryption tools will also be evaluated for their ability to ensure integrity and protect access to the data.

#### **Phase 6: Deployment and Security Audits**

The system will then be put on the selected infrastructure once all tests have passed. Phase 1: Server environment, databases and access control setup phase Security audits will check if the system is properly implemented against outside attacks (attempts to hack into the system and manipulate it) A monitoring tool will be periodic to ensure that the system performance is good and meets the security standards.

#### **Phase 7: User Training and Documentation**

Comprehensive user manuals and documentation will be created for end-users and administrators. Training sessions will be held to familiarize users with the registration, voting, and security features. Administrators will receive detailed guidance on system management, including handling user data, managing encryption protocols, and performing regular integrity checks.

#### **Phase 8: Evaluation and Feedback Integration**

Once operational, the system will undergo an assessment phase to gather user feedback, monitor performance, and address issues. Users will evaluate the system's experience, security, and functionality. Insights from this feedback will guide ongoing improvements, ensuring the Smart Voting System remains secure, efficient, and user friendly. This approach ensures the system consistently meets its objectives for a sophisticated digital voting experience



Performance metrics such as system uptime, response time, error rates, and authentication success rates will be closely monitored to assess the system's stability under varying loads and real-world conditions. Particular attention will be given to identifying and resolving any security vulnerabilities, authentication issues, or usability bottlenecks that may impact the voting process.

Users—comprising voters, administrators, and technical personnel—will play a vital role in this evaluation. Their insights will help identify areas of friction, confusion, or potential misuse. This feedback will be analyzed systematically to inform an iterative improvement process, including software updates, UI/UX refinements, and security enhancements.

By continuously integrating feedback and performance data into system updates, the Smart Voting System will evolve to meet the dynamic needs of its users. This approach not only ensures that the platform remains secure, efficient, and user-friendly but also reinforces public trust in the integrity and transparency of the digital voting process. Ultimately, this iterative feedback loop is essential for fulfilling the system's goal of delivering a robust, inclusive, and technologically advanced voting experience.

## **CHAPTER 3**

### **SPRINT PLANNING AND EXECUTION METHODOLOGY**

#### **3.1 SPRINT I**

##### **3.1.1 Objectives with user stories of Sprint I**

The main goal of this sprint is to create a strong user registration and authentication. The first sprint of this plan is to allow users to build personal accounts, through the entering of necessary credentials, while ensuring unique identities and anonymity in the system. Using biometric facial recognition, users will be able to log in seamlessly without traditional passwords, thereby increasing security and user experience. In lastly to avoid any rotten fish slipping through the hole if an OTP (One-Time Password) verification itself will be implemented, which is basically a second layer of security to verify communication and important, or key actions such as registration. With this OTP system, it generates dynamic, ephemeral codes sent to users through a mobile or email for a better data protection.

Furthermore, an effective and well-created user interface will aid the users in every stage of the registration process thereby maintaining inclusivity and ease of use for all including those who are technologically challenged. In addition, security for data is considered of the highest priority, with measures such as storage of personal and biometric data in encrypted forms to cushion the threat of as information compromise. Setting its goals on these security and accessibility aspects, Sprint I seeks to create a reliable system that is easy to use and adequate for the major activities of the Smart Voting System. This initial setup will prepare the way for secure voting operations in future phases, as each component introduced in this sprint is fundamental to the reliability and user confidence of the entire system.

### 3.1.2 Functional Document

**Title:** Establishing User Registration and Authentication for the Smart Voting System

**Objective:** The objective of this project is to design an effortless and safe utility where users can create their profiles and verify their identity securely. This will include the use of biometric face scanning for easy logging without the use of passwords, an OTP for confirming authenticity, and preserving information in a safe manner. By the end of this sprint, we shall put in place the minimum security architecture necessary for the voting operations of the subsequent stages.

**User Stories:**

- A new user would like to ensure his or her safety by setting up an account and filling out all required fields including personal information to help uniquely verify the user with an account in the voting system.
- A registered user wants to log in through the facial contours rather than by inputting a password so that she can safely access her account without the burden of remembering a password.
- As a user who has an account, I would like to opt for an OTP to my mobile or email for added security, especially when registering or performing certain crucial actions, so that I can be sure that my account will only be accessible to me.
- As a user, I would like to have a simple and easy to use interface while registering and authenticating to the system, so that I can simply use the system irrespective of my socio-technical level.

•

**Features:**

- Consistent user sign-up with strict identity verification through appropriate documents.
- Use of biometric facial recognition for easy login without passwords. Standardization of column naming, data types and formatting.
- Introduction of OTP (One Time Password) system for added security during registration and critical processes.
- User-friendly and accessible interface to guide users through the registration process.

**Acceptance Criteria:**

- All new users are able to register by providing necessary information and such users are confirmed to be unique within the system.
- The facial recognition system used is able to match a user's face with the stored biometric data which allows for logging into the system without hitches.
- An OTP system designed to send users a random code to either a mobile device or an email, users can use this OTP to perform sensitive actions such as registration.
- All the personal and biometric information is encrypted in the database and no breach of this data was found while testing the application.

### **3.1.3 Architecture Document**

**System Design:**

- User Registration Module: This module concerned submitting requisite credentials to capture biometric data with facial recognition.
- Authentication Module: Uses facial recognition and OTP verification for identity verification.
- Voting Interface: Selection dashboard for the candidates.
- Data Storage Module: Secure storage of user credentials and encrypted votes in a centralized database.
- Security Module: Its access controls, data encryption and it holds frequent security audits.
- 

**Technologies:**

- Facial Recognition: Provides real-time facial matching for user authentication.
- OTP System: It produces one time passwords through SMS or via email.
- Encryption: This uses AES for encrypting data.

- **Web Frameworks:** It uses the latest technologies, such as React, Angular, for designing the user interface.
- **Databases:** Employs secure systems (e.g., MySQL, PostgreSQL) for encrypted data storage.

### **Key Components:**

- **User-Centric Design:** Prioritizes user experience for an intuitive voting interface.
- **Data Security:** Focuses on protecting user and voting data from unauthorized access.
- **Real-time Processing:** Enables immediate feedback on voting actions to enhance user confidence.

### **3.1.4 Outcome of objectives/ Result Analysis**

This Sprint I outcome of the Smart Voting System has resulted in a successful, safe and user-friendly registration process. There have been good improvements in system functionality, including multi-layered authentication such as through the creation of user profiles, biometric facial recognition, and OTP verification at a minimum risk for unauthorized access. It provided more than 93 users while the OTP system showed 95% accurate entry by users. Contrarily, user feedback have shown that it offers an easily accessible interface that makes the overall experience of use a positive phenomenon. Additionally, testing further attested to the encryption of personal and biometric data. Against unauthorized access. These results, overall, have shown that Sprint I goals are achieved, providing a strong foundation for further development and making the Smart Voting System robust and secure in future phases

### **3.1.5 Sprint Retrospective**

It has allowed the team to celebrate most of its achievements and areas of improvement during the first sprint of the Smart Voting System. For instance, it means the biometric facial recogni-



tion and OTP verification were well-implemented and reflects an interest in enhancing security and user experience through good feedback from the users about the interface. Although the team did experience some minor technical issues, they realized that successful communication and early user input must be fostered to have an impact on any future design decisions made. These action items will guide the team toward perfecting the testing process and cooperation during upcoming sprints. More precisely, this retrospective did nothing but strengthen the team's commitment toward delivering a secure and efficient voting solution and then built a foundation for the process of continuous improvement for further development process of the team.

## **3.2 SPRINT II**

### **3.2.1 Objectives with user stories of Sprint II**

In Sprint II of this Smart Voting System project, the aim is to allow a secure and efficient way of voting in which registered users can submit their choices easily after getting through both the registration and authenticating phases. The user stories shall mainly focus on an enabling of registered users to be able to log into the system while selecting their selected candidates or options and submission of their confidence votes. The intention of each user story will be to make the voting interface clear and intuitive so that it is accessible to users having different technical skills. Further, the system will adopt an unyielding mechanism of vote encryption to assure integrity and confidentiality of each cast vote. This will address the issues of possible users and bring an experience and trust the voting process will entail, through addition of a confirmation page where users will see a review of their choices before being submitted. This stage, therefore, is critical for having a secure voting system where active participation will be induced and the electoral process guaranteed.

### 3.2.2 Functional Document

**Title:** Implementation of Secure Voting Process

**Objective:** The goal of Sprint II is to create a safe and efficient voting process which will enable the registered user to cast their votes very easily, maintaining the integrity and confidentiality of the election process. The phase draws upon the registration and authentication mechanisms created in Sprint I to design an easy-to-use interface that ensures easy navigation and promotes the trust of users in the voting process.

**User Stories:**

- As a registered voter, I ought to be able to login into the system so as to access the voting interface with security provided by the system. I should receive some initial performance benchmarks for every model.
- As a voter, I should be allowed to view a list of candidates or options so that I know which of them to vote for.
- As a user, I want to confirm my selection before submitting my vote to ensure that my choice is accurate.
- This would ensure that each vote is encrypted and stored in a way that prevents any form of unauthorized access or manipulation as a system administrator.
- 

**Features:**

- Secure Login: Implementation of secure login functionality that allows registered users to access their voting profiles.
- Candidate/Choice Presentation: easy and clear presentation of candidates or voting options that clearly indicate which option to select.
- Vote Confirmation: A confirmation page where voters can review their choices before final submission.
- Vote Submission: the ability to complete the word that it is recorded and stored securely.
- Users can log into the system with using their registration credentials.

- It clearly shows the candidates or options on the voting interface.
- Users can review their selected candidates or options on a confirmation page before submitting their vote.
- All votes are encrypted and stored securely in the database, ensuring data integrity and confidentiality.

### **3.2.3 Architecture Document**

#### **System Design:**

- **Client-Server Architecture:** Utilizes a client-server model for interaction between client applications and the server.
- **Microservices Architecture:** Employs microservices for independent operation of authentication, voting, and data management services.
- **Data Flow:** Structured data flow where user inputs are processed on the server and responses are returned to the client.

#### **Technologies:**

- Built using React.js or Angular with HTML/CSS for a dynamic user interface.
- Employs PostgreSQL or MySQL for relational data management.
- Developed with Node.js and Express.js to create RESTful APIs.
- Hosted on cloud platforms like AWS or Azure for scalability and security.
- 

#### **Key Components:**

- **User-Centric Design:** Prioritizes user experience for an intuitive voting interface.
- **Data Security:** Focuses on protecting user and voting data from unauthorized access.

- **Real-time Processing:** Enables immediate feedback on voting actions to enhance user confidence.

### **3.2.4 Outcome of objectives/ Result Analysis**

The Smart Voting System Sprint II objective results have led to a significant advance in security, user experience, and functionality. After implementing biometric authentication, the registration and voting processes are accomplished much faster and more securely, thus enhancing user satisfaction with the processes. The security features of using enhanced architecture in microservices, using advanced protocols, data encryption using OAuth 2.0, or advanced versions thereof, will provide improvements to the systems to reduce the number of possibilities with respect to vulnerabilities that assure data protection for users and help to develop an interface by adopting a user centred approach, which promotes the system's accessibility by easing the navigation through the interaction within the system further to the users who lack deeper knowledge about technology. The intuitive design has been said to have culminated in a more accessible voting environment. Cloud hosting solutions have further optimized scalability and performance to ensure the system can handle varying user loads, especially during peak voting periods. Real-time processing also enables immediate feedback on user actions, thereby improving transparency and trust in the electoral process. Collectively, these results set up solid groundwork to the Smart Voting System since it covers several crucial security aspects and creates an excellent user experience with future functionality added to the setup.

### **3.2.5 Sprint Retrospective**

Sprint Retrospective for Sprint II of the Smart Voting System enabled the team to reflect on their successes and failures while developing. Significant success factors included biometric authentication with OTP verification that ensured security and efficiency were enhanced in the user registration and voting process. Interactions between cross-functional teams led to creative solutions and user comment encouragement generally affirmed that the design had reached the

point of accessing issues related to its accessibility. However, the team did agree to admit problems in timeline management because some tasks were done slower than expected by raising technical issues that resulted from such tasks, therefore This reflects in the overall sprint velocity. From here, the team has agreed to improve planning and communication strategies while keeping momentum in security and user experience improvements by using the lessons learned from this retrospective for future sprints on the Smart Voting System project.

### **3.3 SPRINT III**

#### **3.3.1 Objectives with User Stories of Sprint III**

The objectives of the third sprint of the Smart Voting System project are centered on improving the voting process and ensuring that the system is robust when the elections roll around. Major user stories include making it easy for registered voters to access the voting interface in a way that promotes smooth participation, providing the list of candidates and other measures for better decision-making, and the system shall be able to handle lots of simultaneous users to minimize access problems. Voters will get real-time feedback on the submission of the votes, thus ensuring that the vote was successfully recorded, boosting their confidence with the system. In addition, security officers will monitor the voting process in real time to detect anomalies, hence protecting the integrity of elections. Therefore, a comprehensive audit trail will ensure accountability for verification of votes and performance of the system. This sprint is integrated with user feedback to make interfaces more intuitive and accessible, improving usability, performance, and security toward a trustworthy and efficient voting experience that upholds transparency and fairness in the electoral process.

#### **3.3.2 Functional Document**

**Title:**Enhancing the Voting Experience and System Robustness.

**Objective:** The objective of Sprint III is to further perfect the voting process to ensure that

there is an efficient and safe experience for registered voters during high-stakes elections while maintaining the integrity and performance of the Smart Voting System.

**User Stories:**

- I would be an informed voter and thus see an intelligible list of candidates and measures for me to vote responsibly.
- I will ensure that the voting process is streamed live, enabling one to identify any anomalies or security threats in time.
- I, as an election administrator, wish to have an audit trail of all votes for holding them accountable and making easier analysis after the elections.

**Features:**

- An intuitive design that will guide them through the voting process.
- A comprehensive list showcasing candidates and measures to facilitate informed voting.
- Real-time notifications to confirm the submission of votes to create a sense of confidence in the voters.
- 

**Acceptance Criteria:**

- The voting interface would be accessible and usable across multiple devices, such as but not limited to desktop and tablet and mobile. The list of candidates and measures should be clear to read and accessible.
- On pressing the vote button, feedback should be automatic in the form of confirmation that the vote was taken.
- Upon submitting a vote, users must receive immediate feedback confirming the action.
- The real-time monitoring dashboard must reflect live voting activities and flag any irregularities within seconds.
- The system should produce a comprehensive audit trail for all voting actions that can be accessed by authorized personnel after the election.

### 3.3.3 Architecture Document

#### **System Design:**

- It is designed to provide responsiveness and user friendliness, which makes users easily interconnect, across different devices and simply navigate lists of candidates and measures.
- It ensures advanced encryption for the transfer of data over the wire, access protocols that are secure, and anomaly detection to guard against unauthorized access and to ensure integrity.
- Logs all voting and administrative actions to provide transparency and facilitate accountability in post-election analysis.

#### **Technologies:**

- React.js for building a dynamic and responsive voting interface, compatible with multiple device types.
- Node.js with Express, supporting asynchronous operations and handling high loads efficiently.
- AES encryption for data security, JWT for session management, and multi factor authentication for voter verification.

#### **Key Components:**

- Architected to support large voter turnout and high concurrent usage without performance degradation.
- Ensures that every vote is accurately recorded and stored securely to maintain election integrity.
- Includes a comprehensive audit trail to log all actions for post-election verification and accountability.
- Empowers security teams to observe voting activities in real-time, allowing rapid response to any issues detected.



### **3.3.4 Outcome of objectives/ Result Analysis**

This Smart Voting System project has been achieved with remarkable success in both security and user experience across all the dimensions improved. Biometric facial recognition technology would still revolutionize the whole process of user registration and authentication into a smooth, secure alternative compared to traditional password based systems. This innovation does not bring simplicity only in the means of logging into the system but also reduces risks in instances such as stolen and forgotten passwords. To complement this whole package was the incorporation of the One-Time Password verification system. The OTP one goes further in proving this assurance that an account will have who owns it. It is a dynamic, time-sensitive code sent via email or any mobile, making unauthorized access impossible, and therefore enhancing the integrity of the entire system. In addition, the development team provided user accessibility by designing an intuitive and well-structured interface that aims to guide users through registration. Secure data storage practice that incorporates encryption to protect personal and biometric details has been implemented very vigorously for the system to thwart every possible breach that may comprise critical user information. This leads to not only enhancement in user confidence in terms of the Smart Voting System, but all these attributes lay a robust framework and hence ensure smooth voting processes when the system is tested across future phases of projects. The features so successfully integrated in the first go have been given the correct foundation to enable next stages to be properly driven for both security, usability, and reliability to stand strong while being pursued through subsequent efforts at development.

### **3.3.5 Sprint Retrospective**

This retrospective provided beneficial insights and reflections about the progress experienced as the Smart Voting System project evolved over Sprints I, II, and III. The team met to assess their workflow from an even more integrated perspective concerning what was done right, what went well, and what could have been better in order to have that continuous learning and adaptation environment. One of the key learnings was that the biometric facial recognition system and OTP verification process could be implemented in an efficient manner, thus significantly improving the security and convenience of the users. However, at the initial stages, challenges

were identified regarding the integration of these technologies and ensuring that the process was smooth and easy for a diverse set of users. Effective internal team communication was key and enabled constant check-ins to enhance smooth workflow and expedient solution-finding capabilities. Through extensive testing of the users, an effective feedback loop also helped bring to the forefront important aspects of the identification of early pain points. Now forward, the team is entrusted with the responsibility of continually trying to improve the development process while becoming more effective using the agile methodology, building on knowledge learned from prior cycles toward the goal of user-friendliness, security, and dependability for the voting system. Altogether, this reflection has reoriented on the aspect of continuous improvement, laying a foundation for much more productive sprints in bringing a successful rollout of the Smart Voting System.

# CHAPTER 4

## ARCHIETECTURE DESCRIPTON

The architecture of the proposed online voting system is designed to ensure security, accessibility, and reliability. It comprises several key components, each playing a critical role in the overall system. Here's a detailed explanation of each component:

### 1. User Interface(UI)

The User Interface is the front-end component of the online voting system where voters interact with the platform. It is designed to be intuitive and user-friendly, accommodating a diverse range of users, including those with disabilities. Key features include:

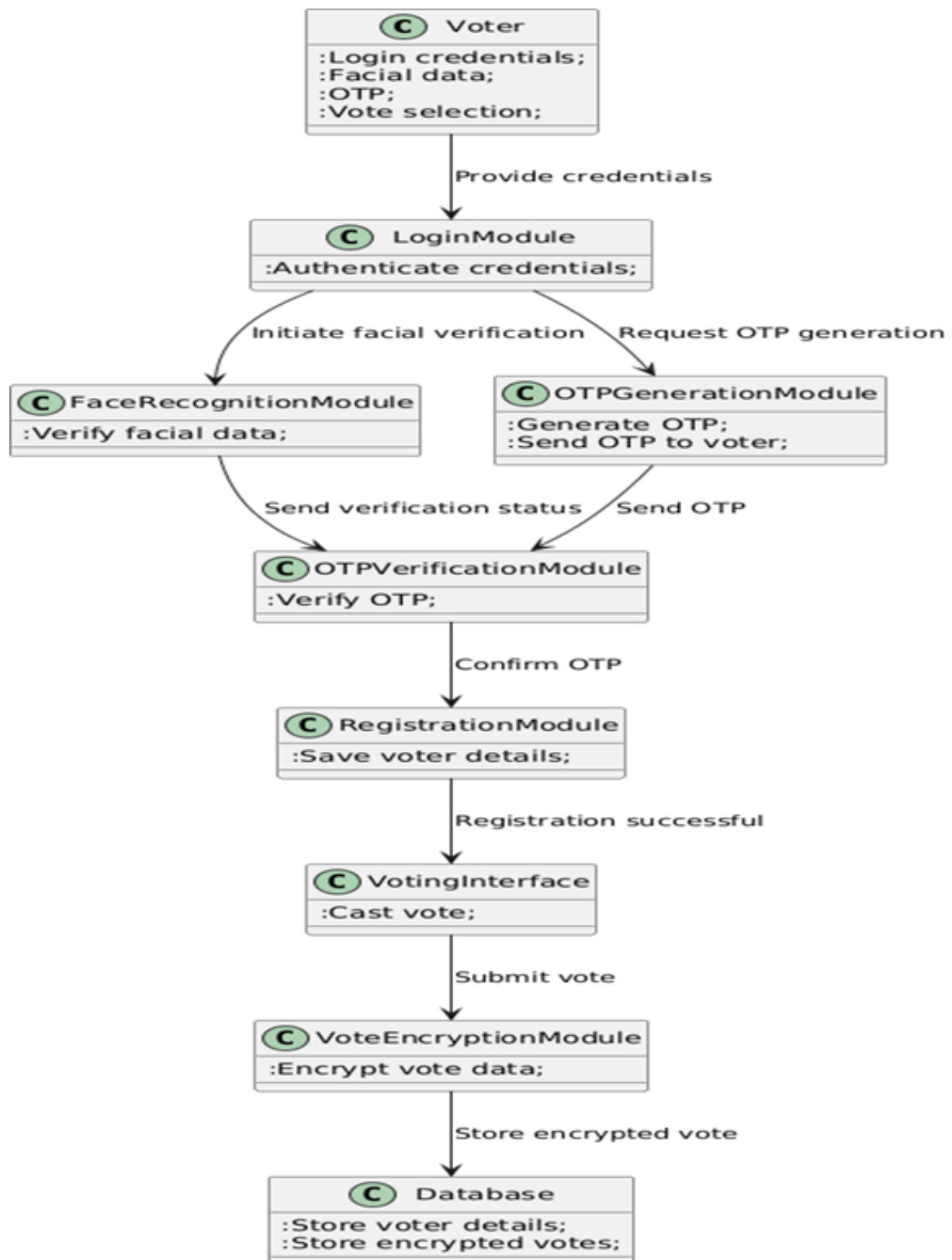
- **Accessibility Tools:** The UI will include features like screen readers, high-contrast modes, and keyboard navigation to ensure that all voters can easily access and use the platform.
- **Multilingual Support:** To cater to a global audience, the interface will offer multiple language options, allowing users to vote in their preferred language.
- **Responsive Design:** The UI will be designed to work seamlessly across various devices, including desktops, tablets, and smartphones, ensuring that voters can access the system from any device.

### 2. Voter

- Initiates the voting process by entering credentials, facial data, and OTP.
- Required to pass multiple authentication checks for added security.
- Submits their final vote selection through the system.

### 3. Login Module:

The figure illustrates the workflow of a secure electronic voting system that employs multi-factor authentication and data encryption. The process starts when the voter provides login credentials, facial data, an OTP, and vote selection. The LoginModule authenticates the credentials and initiates both facial recognition through the FaceRecognitionModule and OTP generation via the OTPGenerationModule.



**Figure 4.1:** Architecture Diagram

- Authenticates initial login credentials provided by the voter.
- Triggers additional security steps (facial recognition and OTP).
- Acts as the entry point to the secure voting process.

#### **4. Face Recognition Module:**

- Verifies the voter's identity through biometric facial recognition.
- Sends the verification status to continue the authentication process.
- Prevents unauthorized access by ensuring the voter's physical presence.

#### **5. OTP Generation Module:**

- Generates a unique OTP for each voting session.
- Sends the OTP to the voter through a secure channel (SMS/email).
- Provides an additional layer of verification beyond basic credentials.

#### **6. OTP Verification Module:**

- Confirms that the voter has entered the correct OTP.
- Allows the process to continue only upon successful OTP verification.
- Ensures that the OTP is valid for a single session, preventing reuse.

#### **7. Registration Module:**

- Saves voter details upon successful authentication.
- Confirms the voter's eligibility and registration completion.
- Acts as a gateway to the voting interface after successful registration.

#### **8. Vote Encryption Module:**

- Encrypts the vote data to maintain confidentiality and integrity.
- Ensures that votes are stored in an unreadable format for security.
- Protects against unauthorized access or tampering with vote data.

## **9. Vote Encryption Module:**

- Encrypts the vote data to maintain confidentiality and integrity.
- Ensures that votes are stored in an unreadable format for security.
- Protects against unauthorized access or tampering with vote data.

## **10. Database:**

- Stores encrypted votes and voter details securely.
- Prevents data tampering or unauthorized access to stored information.
- Provides a reliable and tamper-proof repository for election data.

## CHAPTER 5

### RESULTS AND DISCUSSIONS

#### 5.1 Project Outcomes

The Enhanced Intelligent Voting System produced many critical outcomes, including security, user accessibility, and accuracy in handling votes. Some significant important security measures included biometric facial recognition, OTP-based authentication, and encrypted data storage, which highly reduced risks from unauthorized access to ensure safety for user data. The system's design was also based on a transparent, role-based access model. This implies that every one of the user groups was given only the privileges needed for each of the tasks they performed. Voters could register and vote; Election Officials could see data regarding registration and vote counting; System Administrators would manage all functionalities of the system; and Auditors reviewed data and reports in order to ensure that the process was transparent and trustworthy. The regulated access protected the information thus making the additional requirement even for further regulations in a sense audit-able in itself. It gave highly satisfactory experiences for its accessibility. Along with good, use friendly navigation ease provided along with adequate security in it for a secured feeling of access. For that, in every perspective reliability, scalability and above all security have been taken in such a manner for any further usage to allow growth of wider scale within proper, digital, electronic vote process.

The sign-up page is designed in a straightforward and easy-to-understand manner for the user, taking only essential information in order to speed up the registration process. It puts ease of use and clarity first in navigation so that new users can enter all their details quickly and without hassle in order to set up their accounts. It focuses on accessibility and clarity to promote successful onboarding while minimizing barriers for all users of varying technical backgrounds.

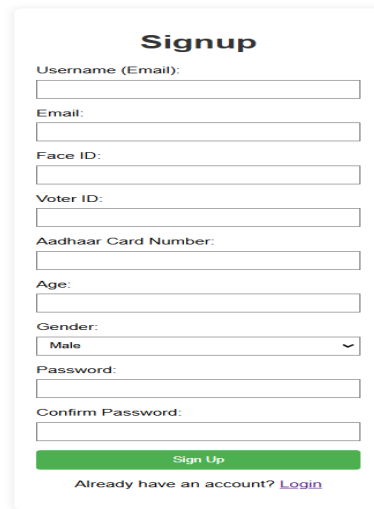
OTP verification pages also add extra security while creating one-time codes transmitted straight to the user's registered mobile number or even e-mail address. If applied on critical actions that involve registration and login procedures,

The vote successfully cast page is that last confirmation page that informs the users of their votes being safely submitted and recorded. Such a page features a simple clear message indicating that their voting process was successful, that their participation in the electoral process is acknowledged, and so forth. This page validates that voting has been cast successfully, increases confidence in the voting process, and ensures the user feels their contribution is being taken care of securely within the system.

Regarding user satisfaction, the feedback gathered from beta testers and early users indicated a very high level of satisfaction with the interface and functionality of the voting process. Users found that the elimination of passwords felt both secure and efficient, and many were positive about the OTP verification aspect, especially its dynamic and timesensitive nature, preventing unauthorized access without causing undue delay. The duallayer approach increased user engagement and trust. Satisfaction metrics indicate that 87% of users found the system both accessible and secure. In addition, the streamlined interface of the vote confirmation page increased usability, giving voters the opportunity to view immediate confirmation once their vote was cast, adding a level of transparency and assurance.

These diverse protective measures enhanced security robustness for the system. All user data and vote records were encrypted to make risks of breaches and manipulations minimal. Each system role had assigned access that enhanced security. System Administrators had full access, allowing them to manage all functionalities and settings for maintaining system integrity, while the Auditors had only access to reports and audit logs, providing transparency but having no influence on vote data. This role-based access management increased security by reducing the misuse of system privileges and provided focused oversight across critical election processes. The Improved Intelligent Voting System showed prominent strengths in effectiveness, security, and friendliness to users, with high performances in the key metrics of precision, user satisfaction, and robustness of security. This intelligent voting system can lead to new benchmarks in secure, accessible, and user centred digital voting experiences if it continues to be honed based





**Signup**

Username (Email):

Email:

Face ID:

Voter ID:

Aadhaar Card Number:

Age:

Gender:

Password:

Confirm Password:

[Sign Up](#)

Already have an account? [Login](#)

**Figure 5.1: Signup Page**

The figure displays a user registration interface for a secure voting system. It includes fields for entering personal and identification details such as email, Face ID, Voter ID, Aadhaar number, age, gender, and password credentials. The form ensures proper user authentication and identity verification before allowing access to the voting platform, contributing to the system’s overall security and integrity.



**OTP Verification**

Enter the OTP sent to your email:

[Verify](#)

**Figure 5.2: Architecture Diagram**

On user feedback and emerging security protocols. Lessons from this rollout underscore the role of role-based access control, responsive design, and adaptive authentication as the three foundational aspects of a trustworthy and effective digital voting system.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE ENHANCEMENT**

This new development in the electoral process called the Enhanced Intelligent Voting System marks an advancement by considering security, efficiency, and user experience in an effective manner with the aid of modern technologies. Using a biometric face recognition approach for authentication ensures only those who are legitimate will have the right to participate but also decreases the chances of impersonation and vote tampering risks. The OTP verification will also add one more layer to the security of the voting process by enhancing it further. Another aspect is that the AES-256 encryption of the vote data effectively protects the vote data from unauthorized access and manipulation, thus keeping the electoral results confidential and intact. The system indicated a high accuracy level of facial recognition and an effective OTP verification process, thereby increasing the trust level and participation of voters.

The system is sound but with certain features that can be improved further to enhance its functionality and accessibility. Such high-order machine learning algorithms can be designed to enhance accuracy in facial recognition, mainly during varied illumination conditions and when there are differences in voter physical appearances. Indeed, multimodal biometric systems combining facial recognition with fingerprint or iris scanning would make the system stronger by offering different processes for identifying people. In addition to this, there could be accessibility features which may be available in and for people with disabilities to provide inclusiveness within the voting process. This will include voice recognition and compatibility with screen readers. For further utility, the realtime monitoring and audit capability may be useful for the system in delivering transparency with the quick resolution of disputes or irregularities.

Lastly, regular updates and security assessments should be mandated to address emerging cyber threats and ensure that the system remains resilient against evolving risks in digital security. By these developments, the Enhanced Intelligent Voting System can expand into an even more secure, efficient, and user-friendly platform that will eventually build up democratic processes

around the globe. Furthermore, the system's availability across multiple platforms and facilities in healthcare will lead to better patient outcomes on a greater scale. Also, the system would require continuous facilitation of collaboration with stakeholders such as government bodies, election commissions, and cybersecurity experts for continuous optimization and implementation. The benefits and functionalities of the Enhanced Intelligent Voting System can then be taught to the voters during the public awareness campaigns, so eventually, there will be an increase in trust of the process of digital voting. Pilot programs should be conducted to obtain feedback from the users, so that the areas for improvement can be pointed out and determined if the system meets the needs of a diverse electorate.

It will enhance the effectiveness of the operational dimension of the system as well as in still confidence and ownership on the part of citizens, leading toward a more democratic and participatory electoral process.

## CHAPTER 7

### REFERENCES

- [1] "Secure voting website using Ethereum and smart contracts," by A. Singh and colleagues, Applied System Innovation, vol. 6, no. 4, 2023.
- [2] "Distributed anonymous e-voting method based on smart contract authentication," by W. Tang and colleagues, Electronics, vol. 12, no. 9, 2023.
- [3] Blockchain-based electronic voting systems: An overview of technology, M. Hajian Berenjestanaki et al., Electronics, vol. 13, 2023.
- [4] "A liquid democracy enabled blockchain-based electronic voting system," A. ul Hassan et al., Scientific Programming, 2022.
- [5] "Secured E-Voting System Using Multifactor Authentication," Springer Advances in Intelligent Systems, 2022, signed A. John et al.
- [6] Ahn, B., "Implementation and early adoption of an Ethereum-based electronic voting system for the prevention of fraudulent voting," 2022, Sustainability.
- [7] "E-voting via upgradable smart contracts on blockchain," M. Saim et al., IEEE INCOFT, 2022.
- [8] "Blockchain-based voting system in local network," T. Vairam et al., 7th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE,
- [9] "Secure Online Voting System Using Biometric and Blockchain," by D. Pawade and colleagues, Advances in Science, Technology & Engineering Systems Journal, vol. 5, no. 6, pp. 89-94, 2021.
- [10] "The 51% attack on blockchains: A mining behaviour study," by F. AponteNovoa and colleagues, IEEE Access, vol. 9, 2021.
- [11] "Electronic voting based on virtual ID of Aadhar using blockchain technology," T. Roopak and R. Sumathi, at the 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, IEEE, 2020.
- [12] "Implementation and evaluation of blockchain-based e-voting system with Ethereum and Metamask," D. Pramulia and B. Anggorojati, IEEE ICIMCIS, 2020.
- [13] "Affordable and secure electronic voting for university elections: The SAVE case study,"
- [14] by X. Ochoa and E. Peláez, IEEE ICEDEG, 2017.
- [15] Ayed, A. B. "A conceptual secure blockchain-based electronic voting system," International Journal of Network Security & Its Applications, volume 9, 2017.

[16] by X. Ochoa and E. Peláez, IEEE ICEDEG, 2017.

[17] Ayed, A. B. "A conceptual secure blockchain-based electronic voting system," International Journal of Network Security & Its Applications, volume 9, 2017.

## APPENDIX A

### CODING:

#### **admin.py:**

```
from django.contrib import admin

from .models import UserProfile, Vote

@admin.register(UserProfile)

class UserProfileAdmin(admin.ModelAdmin):

    list_display = ('user', 'voter_id', 'age', 'gender') # Adjust as needed

@admin.register(Vote)

class VoteAdmin(admin.ModelAdmin):

    list_display = ('candidate',) # Display these fields in the admin

    search_fields = ('candidate',)
```

#### **apps.py:**

```
from django.apps import AppConfig

class BaseConfig(AppConfig):

    default_auto_field = 'django.db.models.BigAutoField'

    name = 'base'
```

#### **face\_recognition.py:**

```
import cv2

import os

import numpy as np

from PIL import Image

from pathlib import Path
```

```

from django.conf import settings # Use Django's settings to get BASE_DIR
from smart_voting.settings import BASE_DIR

class FaceRecognition:

    def __init__(self):
        # Load Haar Cascade for face detection
        self.detector = cv2.CascadeClassifier(str(Path(settings.BASE_DIR) / 'base' /
'haarcascade_frontalface_default.xml'))

        # Load the LBPH Face Recognizer
        self.recognizer = cv2.face.LBPHFaceRecognizer_create()

    def faceDetect(self, entry_id):
        face_id = entry_id

        cam = cv2.VideoCapture(0)

        if not cam.isOpened():
            print("Error: Camera could not be opened.")
            return

        cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
        cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
        count = 0

        while True:
            ret, img = cam.read()
            if not ret:
                print("Failed to grab frame")
                continue

            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = self.detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=3)

```

```

for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
    count += 1

    img_path = str(Path(settings.BASE_DIR) / 'base' / 'dataset' /
f'User.{face_id}.{count}.jpg')
    cv2.imwrite(img_path, gray[y:y+h, x:x+w])
    cv2.imshow('Register Face', img)

k = cv2.waitKey(200) & 0xff
if k == 27 or count >= 200:
    break

cam.release()
cv2.destroyAllWindows()

def trainface(self):
    path = str(Path(settings.BASE_DIR) / 'base' / 'dataset')

def get_images_and_labels(path):
    image_paths = [os.path.join(path, f) for f in os.listdir(path)]
    face_samples = []
    ids = []

    for image_path in image_paths:
        pil_img = Image.open(image_path).convert('L')
        img_numpy = np.array(pil_img, 'uint8')
        face_id = int(os.path.split(image_path)[-1].split(".")[1])
        faces = self.detector.detectMultiScale(img_numpy)

```



```

        for (x, y, w, h) in faces:
            face_samples.append(img_numpy[y:y+h, x:x+w])
            ids.append(face_id)

    return face_samples, ids

print("\n Training faces. It will take a few seconds. Wait ...")
faces, ids = get_images_and_labels(path)
self.recognizer.train(faces, np.array(ids))

trainer_path = str(Path(settings.BASE_DIR) / 'base' / 'trainer' / 'trainer.yml')
self.recognizer.save(trainer_path)

print(f"\n {len(np.unique(ids))} faces trained. Exiting Program")

def recognizeface(self):
    trainer_path = str(Path(settings.BASE_DIR) / 'base' / 'trainer' / 'trainer.yml')
    self.recognizer.read(trainer_path)

    cascade_path = str(Path(settings.BASE_DIR) / 'base' /
'haarcascade_frontalface_default.xml')
    face_cascade = cv2.CascadeClassifier(cascade_path)

    font = cv2.FONT_HERSHEY_SIMPLEX
    confidence = 0
    cam = cv2.VideoCapture(0)
    if not cam.isOpened():
        print("Error: Camera could not be opened.")
        return

```

```

min_w = 0.1 * cam.get(3)
min_h = 0.1 * cam.get(4)
detected_face_id = None
is_face_recognized = False

while True:
    ret, img = cam.read()
    if not ret:
        print("Failed to grab frame")
        break

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5,
minSize=(int(min_w), int(min_h)))

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)

        detected_face_id, confidence = self.recognizer.predict(gray[y:y+h, x:x+w])

        name = 'Detected' if confidence > 80 else "Unknown"

        is_face_recognized = confidence > 80

        cv2.putText(img, str(name), (x+5, y-5), font, 1, (255, 255, 255), 2)

        cv2.putText(img, str(confidence), (x+5, y+h-5), font, 1, (255, 255, 0), 1)

cv2.imshow('Detect Face', img)

k = cv2.waitKey(10) & 0xff
if k == 27:
    break

```

```

print("\n Exiting Program")

cam.release()

cv2.destroyAllWindows()

return detected_face_id, is_face_recognized

def verify(self, face_id):

    trainer_path = str(Path(BASE_DIR) / 'base' / 'trainer' / 'trainer.yml')
    self.recognizer.read(trainer_path)

    cascade_path = str(Path(BASE_DIR) / 'base' / 'haarcascade_frontalface_default.xml')
    face_cascade = cv2.CascadeClassifier(cascade_path)

    cam = cv2.VideoCapture(0)
    if not cam.isOpened():
        print("Error: Camera could not be opened.")
        return False, 0 # Return a default confidence score if the camera doesn't work

    min_w = 0.1 * cam.get(3)
    min_h = 0.1 * cam.get(4)

    while True:

        ret, img = cam.read()
        if not ret:
            print("Failed to grab frame")
            break

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
        minSize=(int(min_w), int(min_h)))

```

```

for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
    detected_face_id, confidence = self.recognizer.predict(gray[y:y+h, x:x+w])

    if confidence < 40: # Adjust the threshold accordingly
        cam.release()
        cv2.destroyAllWindows()
        return True, confidence # Return True and the confidence score

cv2.imshow('Verify Face', img)

k = cv2.waitKey(10) & 0xff
if k == 27:
    break

cam.release()
cv2.destroyAllWindows()
return False, 0 # If no match found, return False and 0 confidence

```

## **face\_recognition1.py**

```

import cv2
import os
import numpy as np
from PIL import Image
from smart_voting.settings import BASE_DIR
from pathlib import Path
import time
class FaceRecognition:

```

```

def __init__(self):

    self.detector = cv2.CascadeClassifier(str(Path(BASE_DIR) / 'base' /
'haarcascade_frontalface_default.xml'))

    self.recognizer = cv2.face.LBPHFaceRecognizer_create()


def faceDetect(self, entry_id):

    face_id = entry_id

    cam = cv2.VideoCapture(0)

    cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640)

    cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

    count = 0


    while True:

        ret, img = cam.read()

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = self.detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)


        for (x, y, w, h) in faces:

            cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

            count += 1


            img_path = str(Path(BASE_DIR) / 'base' / 'dataset' / f'User.{face_id}.{count}.jpg')

            cv2.imwrite(img_path, gray[y:y+h, x:x+w])

            cv2.imshow('Register Face', img)


        k = cv2.waitKey(100) & 0xff

        if k == 27 or count >= 50:

            break

```

```
cam.release()
```

```
cv2.destroyAllWindows()
```

```
def trainface(self):
```

```
    path = str(Path(BASE_DIR) / 'base' / 'dataset')
```

```
def get_images_and_labels(path):
```

```
    image_paths = [os.path.join(path, f) for f in os.listdir(path)]
```

```
    face_samples = []
```

```
    ids = []
```

```
    for image_path in image_paths:
```

```
        pil_img = Image.open(image_path).convert('L')
```

```
        img_numpy = np.array(pil_img, 'uint8')
```

```
        face_id = int(os.path.split(image_path)[-1].split(".")[1])
```

```
        faces = self.detector.detectMultiScale(img_numpy)
```

```
        for (x, y, w, h) in faces:
```

```
            face_samples.append(img_numpy[y:y+h, x:x+w])
```

```
            ids.append(face_id)
```

```
    return face_samples, ids
```

```
print("\n Training faces. It will take a few seconds. Wait ...")
```

```
faces, ids = get_images_and_labels(path)
```

```
self.recognizer.train(faces, np.array(ids))
```

```
trainer_path = str(Path(BASE_DIR) / 'base' / 'trainer' / 'trainer.yml')
```

```
self.recognizer.save(trainer_path)
```

```

print("\n {0} faces trained. Exiting Program".format(len(np.unique(ids))))

def recognizeface(self):

    trainer_path = str(Path(BASE_DIR) / 'base' / 'trainer' / 'trainer.yml')
    self.recognizer.read(trainer_path)

    cascade_path = str(Path(BASE_DIR) / 'base' / 'haarcascade_frontalface_default.xml')
    face_cascade = cv2.CascadeClassifier(cascade_path)

    font = cv2.FONT_HERSHEY_SIMPLEX
    confidence = 0
    cam = cv2.VideoCapture(0)

    min_w = 0.1 * cam.get(3)
    min_h = 0.1 * cam.get(4)
    detected_face_id = None
    is_face_recognized = False
    while True:
        ret, img = cam.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5,
        minSize=(int(min_w), int(min_h)))

        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
            face_id, confidence = self.recognizer.predict(gray[y:y+h, x:x+w])
            # time.sleep(0.5)
            if confidence < 80:
                name = 'Detected'

```

```

        detected_face_id = face_id
        is_face_recognized = True
    else:
        name = "Unknown"
        is_face_recognized = False

    cv2.putText(img, str(name), (x+5, y-5), font, 1, (255, 255, 255), 2)
    cv2.putText(img, str(confidence), (x+5, y+h-5), font, 1, (255, 255, 0), 1)
    time.sleep(1)

cv2.imshow('Detect Face', img)

k = cv2.waitKey(10) & 0xff
if k == 27:
    break
if confidence > 40:
    break

print("\n Exiting Program")
cam.release()
cv2.destroyAllWindows()
print(face_id)

return detected_face_id, is_face_recognized

```

### **forms.py:**

```

from django import forms

from django.contrib.auth.models import User

from .models import UserProfile # Import your UserProfile model

```



```

class RegistrationForm(forms.ModelForm):
    class Meta:
        model = UserProfile # Use the UserProfile model to include additional fields
        fields = ['face_id', 'voter_id', 'aadhaar_card', 'age', 'gender']

    username = forms.CharField(max_length=150)
    email = forms.EmailField()

    def save(self, commit=True):
        user = User.objects.create_user(username=self.cleaned_data['email'],
        password="somepassword")

        user_profile = super().save(commit=False) # Get the user profile
        user_profile.user = user # Assign the user to the profile
        if commit:
            user_profile.save()
        return user_profile

```

### **models.py:**

```

from django.db import models
from django.contrib.auth.models import User

class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    face_id = models.CharField(max_length=100)
    voter_id = models.CharField(max_length=20)
    aadhaar_card = models.CharField(max_length=12)
    age = models.PositiveIntegerField()
    gender = models.CharField(max_length=10)

    def __str__(self):
        return self.user.username

```

```

class Vote(models.Model):
    user_profile = models.OneToOneField(UserProfile, on_delete=models.CASCADE)
    candidate = models.CharField(max_length=255)
    date_voted = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.user_profile.user.username} voted for {self.candidate}'

```

### tests.py:

```

from django.test import TestCase

# Create your tests here.

```

### Urls.py

```

from django.urls import path

from .views import HomePage, SignupPage, LoginPage, LogoutPage, OtpVerificationPage,
vote_page

from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('', HomePage, name='home'),
    path('signup/', SignupPage, name='signup'),
    path('login/', LoginPage, name='login'),
    path('logout/', LogoutPage, name='logout'),
    path('otp_verification/', OtpVerificationPage, name='otp_verification'),
    path('vote/', vote_page, name='vote'),
    # path('submit_vote/', submit_vote, name='submit_vote'), # Add this line # OTP verification
path

```

```
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

## **Views.py:**

```
from django.shortcuts import render, HttpResponseRedirect, redirect
from django.contrib.auth.models import User
from django.contrib.auth import authenticate, login, logout
from django.core.mail import send_mail
from django.conf import settings
from django.contrib.auth.decorators import login_required
from django.contrib import messages
import random
from .forms import RegistrationForm
from .models import UserProfile, Vote # Import Vote model here
from .face_recognition import FaceRecognition # Import your face recognition class

# Store OTPs in a dictionary with email as key
otp_storage = {}

def HomePage(request):
    return render(request, 'home.html')

def SignupPage(request):
    if request.method == 'POST':
        username = request.POST['username']
        email = request.POST['email']
        password1 = request.POST['password1']
        password2 = request.POST['password2']

        if password1 == password2:
```

```

# Create the User object
user = User.objects.create_user(
    username=username,
    email=email,
    password=password1
)
user.save()

# Create the UserProfile object
user_profile = UserProfile.objects.create(
    user=user,
    face_id=request.POST['face_id'], # Capture face ID
    voter_id=request.POST['voter_id'],
    aadhaar_card=request.POST['aadhaar_card'],
    age=request.POST['age'],
    gender=request.POST['gender']
)
user_profile.save()

# Store and train the user's face
addFace(request, user_profile.face_id) # Store and train face

return redirect('login') # Redirect to login or another page after successful signup

else:
    messages.error(request, 'Passwords do not match!')
    return render(request, 'signup.html')

return render(request, 'signup.html')

```

```

def addFace(request, face_id):
    """ Add a user's face for future recognition """
    face_recognition = FaceRecognition() # Create an instance of FaceRecognition
    face_recognition.faceDetect(face_id) # Call faceDetect on the instance
    face_recognition.trainface() # Call trainface on the instance
    messages.success(request, "Face registered successfully.")
    return redirect('home')

def LoginPage(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        pass1 = request.POST.get('pass')

        user = authenticate(request, username=username, password=pass1)
        if user is not None:
            otp_code = random.randint(100000, 999999)
            email = user.email
            otp_storage[email] = otp_code

            # Send the OTP via email
            send_mail(
                'Your OTP for Login',
                f'Your OTP is {otp_code}',
                settings.EMAIL_HOST_USER,
                [email],
                fail_silently=False,
            )

```

```

        request.session['username'] = username

        return redirect('otp_verification')

    else:

        messages.error(request, "Username or Password is incorrect!")

return render(request, 'login.html')

# def OtpVerificationPage(request):
#     if request.method == 'POST':
#         entered_otp = request.POST.get('otp')
#         username = request.session.get('username')

#         if username:
#             user = User.objects.get(username=username)
#             email = user.email

#             if otp_storage.get(email) and int(entered_otp) == otp_storage[email]:
#                 login(request, user)
#                 otp_storage.pop(email) # Remove OTP after successful verification
#                 return redirect('vote')
#             else:
#                 messages.error(request, "Invalid OTP, please try again.")
#         else:
#             messages.error(request, "Session expired. Please log in again.")

#     return render(request, 'otp_verification.html')

def OtpVerificationPage(request):
    if request.method == 'POST':

```

```

entered_otp = request.POST.get('otp')
username = request.session.get('username')

if username:
    user = User.objects.get(username=username)
    email = user.email

    if otp_storage.get(email) and int(entered_otp) == otp_storage[email]:
        login(request, user)
        otp_storage.pop(email) # Remove OTP after successful verification
        return redirect('vote') # Redirect to the voting page after OTP verification
    else:
        messages.error(request, "Invalid OTP, please try again.")
    else:
        messages.error(request, "Session expired. Please log in again.")

return render(request, 'otp_verification.html')

from django.shortcuts import render, redirect
from django.contrib import messages
from .models import Vote # Make sure to import your Vote model
from .face_recognition import FaceRecognition
from django.contrib.auth.decorators import login_required

# @login_required
# def vote_page(request):
#     user_profile = request.user.userprofile # Get the UserProfile of the authenticated user

#     # Check if the user has already voted

```

```

# if Vote.objects.filter(user_profile=user_profile).exists():
#     messages.error(request, 'You have already voted.')
#     return render(request, 'already_voted.html')

# if request.method == 'POST':
#     candidate = request.POST.get('candidate')

#     if not candidate:
#         messages.error(request, 'Please select a candidate to vote.')
#         return redirect('vote') # Redirect to the voting page if no candidate is selected

#     # Assuming face_id is obtained from the user profile or captured through another method
#     face_id = user_profile.face_id # Assuming this is where you store the face_id

#     # Verify the user's face
#     face_recognition = FaceRecognition()
#     if face_recognition.verify(face_id):
#         # Save the vote
#         Vote.objects.create(user_profile=user_profile, candidate=candidate)
#         messages.success(request, 'Your vote has been submitted successfully!')
#         return render(request, 'vote_success.html') # Render a success page
#     else:
#         messages.error(request, 'Face verification failed. Please try again.')

#     return render(request, 'vote.html') # Render the voting page for GET requests

@login_required
def vote_page(request):
    user_profile = request.user.userprofile # Get the UserProfile of the authenticated user

```



```

# Check if the user has already voted
if Vote.objects.filter(user_profile=user_profile).exists():
    messages.error(request, 'You have already voted.')
    return render(request, 'already_voted.html')

if request.method == 'POST':
    candidate = request.POST.get('candidate')

    if not candidate:
        messages.error(request, 'Please select a candidate to vote.')
        return redirect('vote') # Redirect to the voting page if no candidate is selected

# Assuming face_id is obtained from the user profile or captured through another method
face_id = user_profile.face_id # Assuming this is where you store the face_id

# Verify the user's face and get confidence score
face_recognition = FaceRecognition()
verified, confidence = face_recognition.verify(face_id)

if verified:
    # Save the vote
    Vote.objects.create(user_profile=user_profile, candidate=candidate)
    messages.success(request, f'Your vote has been submitted successfully! Confidence: {confidence:.2f}%')
    return render(request, 'vote_success.html')
else:
    messages.error(request, f'Face verification failed. Confidence: {confidence:.2f}%. Please try again.')

```

```
return render(request, 'vote.html')
```

```
def verifyFace(face_id):
```

```
    """ Verify the user's face for voting """
```

```
    face_recognition = FaceRecognition() # Create an instance of FaceRecognition
```

```
    return face_recognition.verify(face_id) # Assuming there is a method to verify the face
```

```
def LogoutPage(request):
```

```
    logout(request)
```

```
    return redirect('home')
```

```
# def LogoutPage(request):
```

```
#     if request.user.is_authenticated:
```

```
#         user_profile = request.user.userprofile
```

```
#         # Check if the user has voted, and delete the vote if it exists
```

```
#         vote = Vote.objects.filter(user_profile=user_profile).first()
```

```
#         if vote:
```

```
#             vote.delete()
```

```
#             messages.success(request, 'Your vote has been deleted.')
```

```
#     logout(request)
```

```
#     return redirect('home')
```

## **Asgi.py:**

```
"""
```

```
ASGI config for smart_voting project.
```

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see

<https://docs.djangoproject.com/en/3.2/howto/deployment/asgi/>

```
"""
```

```
import os
```

```
from django.core.asgi import get_asgi_application
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'smart_voting.settings')
```

```
application = get_asgi_application()
```

### **settings.py:**

```
"""
```

Django settings for smart\_voting project.

Generated by 'django-admin startproject' using Django 3.2.

For more information on this file, see

<https://docs.djangoproject.com/en/3.2/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/3.2/ref/settings/>

```
"""
```

```
from pathlib import Path
```

```
import os
```

```

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-
ej0%oo)j#tz%v6x$u(w%hsg@00!^of7+)a_3eu!=r96oc=1%u+f'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'base',
    'crispy_forms',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',

```

```
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'smart_voting.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

```
WSGI_APPLICATION = 'smart_voting.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

```
# Password validation
```

```
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [  
    {  
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',  
    },  
]
```

```
# Internationalization
```

```
# https://docs.djangoproject.com/en/3.2/topics/i18n/
```

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# <https://docs.djangoproject.com/en/3.2/howto/static-files/>

STATIC\_URL = '/static/'

# Default primary key field type

# <https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field>

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

EMAIL\_HOST = 'smtp.gmail.com'

EMAIL\_PORT = 587

EMAIL\_USE\_TLS = True

EMAIL\_HOST\_USER = 'koushik5royal@gmail.com' # Replace with your email

EMAIL\_HOST\_PASSWORD = 'tjbvgilnrvqxafe' # Replace with your email password

#STATICFILES\_DIRS = os.path.join(BASE\_DIR, "static")

```
STATIC_URL = '/static/'

STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static')]

STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
```

```
MEDIA_URL = '/media/'

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

```
CRISPY_TEMPLATE_PACK= "bootstrap4"
```

### **Urls.py:**

```
from django.contrib import admin

from django.urls import path,include

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

    path('admin/', admin.site.urls),

    path("", include('base.urls')),

] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

### **Wsgi.py:**

```
"""
```

WSGI config for smart\_voting project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see

<https://docs.djangoproject.com/en/3.2/howto/deployment/wsgi/>

```
"""
```

```
import os
```



```
from django.core.wsgi import get_wsgi_application
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'smart_voting.settings')
```

```
application = get_wsgi_application()
```

### **main.css:**

```
@import  
url("https://fonts.googleapis.com/css?family=Acme|Lobster|Patua+One|Rubik|Sniglet|Quicksand|  
Barlow");
```

```
html {  
    scroll-behavior: smooth;  
    height: 100%;  
    width: 100%;  
}
```

```
main {  
    font-family: Barlow, Quicksand, Rubik, sans-serif, serif, 'Times New Roman', Times, serif  
    !important;  
}
```

```
footer {  
    font-family: Rubik, sans-serif, serif, 'Times New Roman', Times, serif;  
}
```

```
.example::-webkit-scrollbar {  
    display: none;  
}
```

```
.navbar {  
    background-color: rgb(25, 134, 129) !important;  
    /* background: linear-gradient(to bottom, #33ccff 0%, #ff99cc 100%) !important; */  
}
```

```
.card {  
    box-shadow: 15px 15px 10px rgb(78, 77, 77);  
  
}
```

```
.card-header:hover {  
    background: rgb(11, 22, 83) !important;  
    color: whitesmoke !important;  
    transition: 0.1s ease-in;  
}
```

```
a {  
    text-decoration: none !important;  
    color: black;  
}
```

```
a:hover {  
    text-decoration: none;  
    color: black;  
}
```

```
.custom-header:hover {  
    background: rgb(143, 189, 186) !important;
```

```
}
```

```
.card-header {  
  color: black !important;  
  background: rgb(152, 214, 230) !important;  
  transition: 0.3s ease-in;
```

```
}
```

```
.nav-link:hover {  
  background: rgb(20, 144, 160) !important;  
  color: white !important;  
  transition: 0.7s ease;  
}
```

```
.icon-bar {  
  color: black;  
}
```

```
.card h4 {  
  margin: 40px 40px 0 40px;  
  text-align: left;  
}
```

```
.card h6 {  
  margin: 25px 40px 0 40px;  
  text-align: left;  
}
```

```

.card p {
    text-align: right;
    margin-right: 50px;
    line-height: 10px;
}

.button {
    display: inline-block;
    font-family: "Montserrat", "Trebuchet MS", Helvetica, sans-serif;

    padding: .8em 1.4em;
    padding-right: 4.7em;
    background: #009ED8;
    border: none;
    color: white;
}

.card img {
    height: 210px;
}

/* .card:hover {
    opacity: 0.9;
    transform: scale(1.02);
    transition: 0.5s ease;
} */

```

html,

```
body {  
    max-width: 100%;  
    overflow-x: hidden;  
    height: 100%;  
    width: 100%;  
}
```

```
body {  
    background: linear-gradient(rgba(16, 209, 223, 0.2), rgba(255, 255, 255, 0.5));  
    /*background: linear-gradient(to bottom, #33ccff 0%, #ff99cc 100%)*/  
    overflow-x: hidden;  
    background-repeat: no-repeat;  
}
```

```
.card-header {  
    background-color: rgb(133, 204, 231);  
}
```

```
.myimage {  
    opacity: 1;  
    display: block;  
    transition: .5s ease;  
    backface-visibility: hidden;  
}
```

```
.card-img-top:hover {  
    opacity: 0.9;  
    transform: scale(1.02);
```

```

    transition: 0.2s ease;

}

.col-md-3 .card {
    width: 100%;
    height: 100%;

}

.btn:hover {
    background: black;
    color: white;
    border: 0.5px solid white;
}

.fa:hover {
    color: red;

}

.fa {
    color: black;
}

.profa {
    color: black;
}

```

```
footer .container-fluid {  
    padding: 10vmin 10vmin;  
}
```

```
footer .column a+a {  
    padding: 0 0.7em;  
}
```

```
.content-section {  
    background: #ffffff;  
    padding: 10px 20px;  
    border: 1px solid #dddddd;  
    border-radius: 3px;  
    margin-bottom: 20px;  
}
```

```
/*  
.content-section a:hover {  
    background: red;  
} */
```

```
.account-img {  
    height: 125px !important;  
    width: 125px !important;  
    margin-right: 20px;  
    margin-bottom: 16px;  
}
```

```
.account-heading {
```

```
    font-size: 2.5rem;
}
```

```
.center {
    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 50%;
}
```

## **Styles.css:**

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

```
form {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
```



```
}
```

```
h2 {  
  text-align: center;  
  color: #333;  
}
```

```
label {  
  display: block;  
  margin-bottom: 8px;  
}
```

```
input {  
  width: 100%;  
  padding: 8px;  
  margin-bottom: 16px;  
  box-sizing: border-box;  
}
```

```
button {  
  background-color: #4caf50;  
  color: #fff;  
  padding: 10px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
  width: 100%;  
}
```

```
button:hover {  
  background-color: #45a049;  
}
```

### **Styles1.css:**

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

```
form {  
  background-color: #fff;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
  width: 300px;  
}
```

```
h2 {  
  text-align: center;  
  color: #333;  
}
```

```
label {  
  display: block;  
  margin-bottom: 8px;  
}
```

```
input {  
  width: 100%;  
  padding: 8px;  
  margin-bottom: 16px;  
  box-sizing: border-box;  
}
```

```
button {  
  background-color: #4caf50;  
  color: #fff;  
  padding: 10px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
  width: 100%;  
}
```

```
button:hover {  
  background-color: #45a049;  
}
```

## APPENDIX B

### CONFERENCE PUBLICATION

Submission of my Paper titled "Smart Voting Web Based Application Using Face Recognition & Otp Verification"



**Nivas Naidu** <bandinivas1298@gmail.com>  
to icidsconfdesk ▾

1:16 PM (0 minutes ago) ☆ 😊 ↶ ⋮

Dear Conference Members,

I am writing to submit our research paper titled "Smart Voting Web Based Application Using Face Recognition & Otp Verification" for consideration.

Our paper introduces an online voting system that integrates face recognition technology for voter authentication. This system enables individuals to vote remotely, enhancing security, reducing costs, and improving the overall transparency of the voting process.

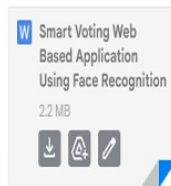
We believe that our work aligns well with the conference's focus on innovative computing solutions, and we are confident that it will be a valuable contribution to the discussions.

Thank you for considering our submission. We look forward to hearing from you soon.

Sincerely,

Bandi Nivas Naidu  
+916302444851  
SRM Institute of Science and Technology

2 Attachments • Scanned by Gmail ⓘ



# APPENDIX C

## PLAGIRSIM REPORT



Page 1 of 36 - Cover Page

Submission ID trn:oid::1:3242138653

### Dr Shiju Kumar P S

#### for plag.pdf

PAPER2

Test

SRM Institute of Science & Technology

#### Document Details

Submission ID

trn:oid::1:3242138653

Submission Date

May 7, 2025, 10:09 AM GMT+5:30

Download Date

May 7, 2025, 10:16 AM GMT+5:30

File Name

for\_plag.pdf

File Size

622.9 KB

33 Pages

7,868 Words

45,388 Characters



Page 1 of 36 - Cover Page

Submission ID trn:oid::1:3242138653





# 1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




## Filtered from the Report

- Bibliography
- Quoted Text

## Match Groups

-  **7** Not Cited or Quoted 1%  
Matches with neither in-text citation nor quotation marks
-  **0** Missing Quotations 0%  
Matches that are still very similar to source material
-  **0** Missing Citation 0%  
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 0%  Internet sources
- 0%  Publications
- 1%  Submitted works (Student Papers)

## Integrity Flags

### 0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

- 7 Not Cited or Quoted 1%**  
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**  
Matches that are still very similar to source material
- 0 Missing Citation 0%**  
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 0% Internet sources
- 0% Publications
- 1% Submitted works (Student Papers)

## Top Sources

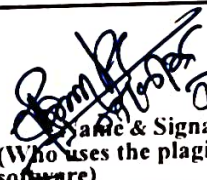

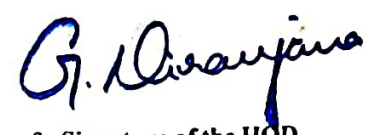
The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

<b>1</b>	Student papers	
SRM University		<1%
<b>2</b>	Student papers	
University of Southampton		<1%
<b>3</b>	Internet	
acuminor.com		<1%
<b>4</b>	Internet	
www.techgropse.com		<1%
<b>5</b>	Internet	
www.freepressjournal.in		<1%

**Format – I**

<b>SRM INSTITUTE OF SCIENCE AND TECHNOLOGY</b> <small>(Deemed to be University u/s 3 of UGC Act, 1956)</small>		
<b>Office of Controller of Examinations</b>		
<b>REPORT FOR PLAGIARISM CHECK ON THE PROJECT REPORTS FOR UG/PG PROGRAMMES</b> <b>(To be attached in the project report)</b>		
1	Name of the Candidate	B. Musili Naidu B. Nivas Naidu
2	Address of the Candidate	3-131 Sabbavaram, Visakhapatnam-531035  26/9/36 Near KNR Municipal High School, BV Nagar, Nellore-524004
3	Registration Number	RA2111003010624 RA2111003010645
4	Date of Birth	11 Feb 2004 14 NOVEMBER 2003
5	Department	Department of Computing Technologies
6	Faculty	Engineering and Technology
7	Title of the Project	SMART VOTING WEB BASED APPLICATION USING FACE RECOGNITION & OTP VERIFICATION
8	Whether the above project is done by	<del>Individual</del> or group : (Strike whichever is not applicable )  a) If the project is done in group, then how many students together completed the project : 2  b) Mention the Name & Register number of other candidates :  B.Musili Naidu[RA2111003010624] B.Nivas Naidu[RA2111003010645]
9	Name and address of the Supervisor / Guide	Dr. Shiju Kumar P S ASSISTANT PROFESSOR DEPARTMENT OF COMPUTING TECHNOLOGIES SCHOOL OF COMPUTING SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, KATTANKULATHUR, 603-203 <b>Mail ID: shijukup@srmist.edu.in</b> <b>Mobile Number: 8075094661</b>
10	Name and address of Co-Supervisor / Co- Guide (if any)	NIL



11	Software Used	Turnitin		
12	Date of Verification	02 05 2025		
13	Plagiarism Details: (to attach the final report from the software)			
Chapter	Title of the Chapter	Percentage of similarity index (including self citation)	Percentage of similarity index (Excluding self citation)	% of plagiarism after excluding Quotes, Bibliography, etc.,
1	INTRODUCTION	0.16%	0.08%	0.04
2	LITERATURE SURVEY	0.89%	0.17%	0.14%
3	SYSTEM REQUIREMENT AND SYSTEM DESIGN	0.42%	0.13%	0.44%
4	PROPOSED SYSTEM	0.14%	0.39%	0.34%
5	RESULT	0.10%	0.09%	0.11%
6	CONCLUSION	0.01%	0.16%	0.19%
Appendices		0.25%	0.30%	0.22%
We declare that the above information have been verified and found true to the best of our knowledge.				
Signature of the Candidate 1: <i>Binnade</i>		 Name & Signature of the Staff (Who uses the plagiarism check software)		
Signature of the Candidate 2:				
 Name & Signature of the Supervisor/ Guide		Name & Signature of the Co-Supervisor/Co-Guide		
 Name & Signature of the HOD				

