Securing Online Voting System Using Visual Cryptography

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Vasu Goel (201435) & Mridul Singhal (201439)

Under the guidance & supervision of

Dr. Diksha Hooda

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "Securing Online voting system using visual cryptography" in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Vasu Goel, 201435" and "Mridul Singhal, 201439" during the period from February 2024 to May 2024 under the supervision of Dr. Diksha Hooda, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Vasu Goel	Mridul Singhal
(201435)	(201439)

The above statement made is correct to the best of my knowledge.

Dr. Diksha Hooda
Assistant Professor (SG)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **'Securing Online Voting System Using Visual Cryptography'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from February 2024 to May 2024 under the supervision of **Dr. Diksha Hooda** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)	(Student Signature with Date)
Vasu Goel	Mridul Singhal
201435	201439

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Dr. Diksha Hooda
Assistant Professor (SG)
Department of Computer Science & Engineering and Information Technology
Dated: May 09, 2024

ACKNOWLEDGEMENT

Firstly, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor Dr. Diksha Hooda, Assistant Professor (SG), Department of CSE/IT, Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of our supervisor in the field of “Information Security” has helped us to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism and valuable advice along with reading many inferior drafts and correcting them at all stage have made it possible for us to complete this project.

We would like to express our heartiest gratitude to Dr. Diksha Hooda, Department of CSE/IT, for her kind help to finish our project.

We would also generously welcome each one of those individuals who have helped us straight forwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated our undertaking.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

Vasu Goel (201435)
Mridul Singhal (201439)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

For free and fair elections to occur, the security of voting systems is crucial. The security of voting systems is frequently threatened by phishing attempts, in which hackers exploit emails or websites to obtain vital data from voters. An approach that shows promise for guarding against phishing attempts on voting systems is visual cryptography. Visual cryptography is where the original message is divided into several parts in this method, and each share is printed on a different transparency. Voters receive the transparencies after which they can only read the original message by carefully superimposing the transparencies to get the image and hence the original message.

The voting system can confirm that the voter is communicating with the authentic system and maintain security by employing visual cryptography. The suggested system offers a reliable and economical way to guarantee voting process security, improving the legitimacy of democratic elections. Attacks involving phishing have advanced significantly, making it challenging to identify and stop them. The fact that voters frequently lack technology expertise makes it difficult to secure voting systems because this leaves them open to phishing attempts. By examining the legitimacy of the transparency materials, they get, voters can also confirm the legitimacy of the system. Third, using visual cryptography offers a strong defense for security since hackers would need to get all of the transparency pieces in order to recover the original message.

Through this model, we purpose to make contributions to the advancement of secure online vote casting practices, providing a framework that no longer most effective strengthens the security of the electoral process but additionally complements person agree with and transparency. The findings and instructions discovered from this endeavor will inform future trends in the area of steady and privateness-keeping on line vote casting structures.

Chapter 1: INTRODUCTION

1.1 INTRODUCTION

With the advancement of technology and growth in IT sector, there is a lot of awareness of online technologies. Considering these growths and awareness, we see many problems in things around us that can be solved or improved with it. One such issue or process is our current paper or ballot paper base voting system. Online voting is a good solution to conduct safe and secure elections. The improved accessibility of internet voting methods is one of their key benefits. Voters no longer need to be present in person at a polling place in order to participate in the electoral process when they cast their ballots online.

There are voters that may be physically disabled or injured at the time of elections, or may belong to some rural place where the accessibility is difficult for the electoral team as well. Online voting is therefore important. Voters can also benefit from the convenience of online voting methods because they do not have to wait in long lines or miss work or other obligations to cast their ballots. Through online voting, we can also count the votes faster and more accurately. This will save both the time, money and energy of election commission. If there won’t be any scope of influence or pressure from any external resource or hacker, this became very good and have best benefits. Online voting systems have benefits, but there are also some drawbacks and difficulties that one may face or must consider. Also, there can be many people who are not aware of how to use the online systems.

There are several benefits of using an online voting system over a conventional paper-based voting system or ballot paper system, making the system more secured, convenient and quicker and give accurate results. To retain the credibility of the electoral process, however, consideration must be given to the security, precision, and accessibility of online voting technologies.

1.2 PROBLEM STATEMENT

There are voting systems in the market that face problems due to the cyberattacks, especially Phishing. These types of attacks trick the users into giving their private and sensitive information like usernames and passwords. Attackers perform and follow many tactics to hack the system including sending spam emails and building fake websites. For example, if a hacker could pose as an election official and send a fake email to the voters asking them to visit a fake voting site to cast their precious votes, the attacker can in
turn manipulate the vote if the fake website managed to steal the information of the voter.

visual cryptography is a concept to protect the system against the phishing attacks performed by hackers in order to make the site more secure. In Visual cryptography we divide an image into several shares, each of which only holds a portion of the original image's information. The original image can only be recreated by combining a particular combination of shares, which can then be distributed to various parties that is we first break the image into n images and hide the text and for decryption, we superimpose all the images into single image so that to get that text back. We can divide each voter's credentials such as usernames and passwords into many shares in order to apply visual cryptography to a voting system. The voter can then receive these shares, each of which can be given by a different means such as email or text message. The voter must combine the shares to reassemble their credentials before they may log in to the voting system. After that, the voter can combine the shares to recreate their ballot, making sure that it is safe and cannot be tampered with by attackers.

Overall, a voting system that employs visual cryptography can offer a reliable approach to defeat phishing attempts and hacker’s attacks and guarantee the accuracy of election outcomes. To make sure the system is secure and confidential, it is crucial to design and implement it thoroughly. To find and fix any problem or vulnerabilities, the system needs to be properly tested. Voters should also receive the appropriate instructions and training which help them with the tutorials to how to vote and can be easy for them to vote without any difficulty.

1.3 OBJECTIVE

The objectives of securing online voting systems using visual cryptography are as follows:

1. The primary objective of this project is to make an effective, user-friendly and seamless user interface (UI) and user experience (UX) for the implementation of “securing online voting system using visual cryptography”.

2. To implement Frontend validations to ensure integrity and security of users input to maintain a smooth and secure voting process that is validating the user’s input for accuracy, completeness and non-repeatable in some stages of input.

3. To add API integration at frontend that is to implement GET and POST requests for getting the data from backend and set the user input including votes and registration page to backend respectively.

4. To add API integration at backend that is to implement GET and POST requests at router end to send the data, add validations, check for security like authentications and other fields.

5. To implement visual cryptography algorithm for enhanced security and to conduct safe and secure elections.

6. To upload the backend on a cloud platform to ensure scalability, reliability and efficient server load management and test the load on server to evaluate the performance of the server and working of APIs

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

To protect the security, integrity and confidentiality of the user, visual cryptography is used to save the voting system from phishing attacks. Elections are very important part for a democratic society so it is very important to conduct them nicely without any external manipulation. If the attackers got unauthorized access to your data, this became very problematic for the users to believe in it as this may compromise the fairness of elections and confidence of public. The motivation is to stop phishing attempts and make the system confidential and secured for the users so that they can vote and choose their representative without any fear.

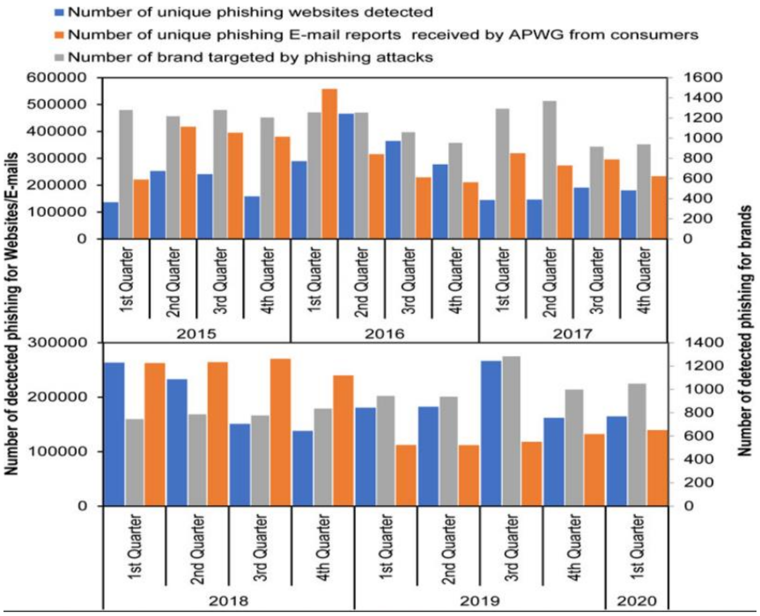


Figure 1.1: Unique phishing E-mails and websites detected (2015-2020)

Our motivation is also taking a part that we will be building a system that is independent and not in influence or pressure of any legal body. Employing visual cryptography to avoid phishing attacks on voting systems is a critical step in ensuring the security, confidentiality, and integrity of the user to vote. Visual cryptography can contribute to greater public confidence in the democratic system and the development of a more representative and democratic society by fostering openness, accessibility, and security in the election process.

1.5 THREE STEP AUTHENTICATION

Three step authentication is a technique where we have three steps of securing the integrity, confidentiality and accessibility of user. This method secures the user 3 times as compare to a single authentication step so that if by any chance Hacker get access or by chance break the one wall of authentication, user would have two more walls left making their identity more secured.

It is far more difficult for hostile actors to compromise user accounts when multiple factors are required for authentication. The extra authentication steps offer an additional layer of protection even in the event that one factor is compromised (e.g., password leaked in a data breach).

The three steps of authentications used in the project “Securing Online voting system using visual cryptography” are:

1. **Email Verification** – This is the first and foremost step of authentication where the admin adds the voter’s email id and the valid user get the mail of the voting link on opening of which they will go to the voting page on successful steps of authentication.

2. **One Time Password (OTP) verification –** Before the start of visual cryptography process user need to go through OTP verification, that is they need to enter their mobile number correctly and will get an OTP on the registered mobile number.

3. **Visual Cryptography –** In visual cryptography, a secret image or message is divided into multiple shares, each of which appears as patterns or random noise. No details about the original image or message can be found in any of these shares alone. On the other hand, the confidential data is revealed when the shares are stacked or overlaid.

1.6 ORGANIZATION OF PROJECT REPORT

CHAPTER 1: INTRODUCTION – The introductory chapter aims to present an overview of the securing of voting system using visual cryptography where first it gives brief introduction about voting system, visual cryptography and then how we will be implementing that in our system. It talks about what the actual problem is and how we are going to solve it, the motivation for doing the project and objectives we will be trying to accomplish during the project.

CHAPTER 2: LITERATURE SURVEY – This chapter aims to present the research papers that we have gone through. What different and new things we have learnt through them. It talks about the overview of all the literature surveys and finally deal with the key gaps that are present in the literature surveys and about their summary.

CHAPTER 3: SYSTEM DEVELOPMENT – The chapter aims to present the both functional and non-functional requirements. It goes through the step-by-step architecture and design of the project. Then it talks about the implementation of the project that include different code snippets, screenshots etc. and finally it gets end with the key challenges that we have faced during the project.

CHAPTER 4: TESTING – This chapter aims to present all the testing we have done in the project from our testing strategy to test cases and outcomes. This section will have all the testing results like validations are working properly or not, database is working correctly or not and many more things like that.

CHAPTER 5: RESULTS AND EVALUATION – This chapter aims to present that the result of all the work we have done. It includes the strength and weaknesses of our project.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE – This chapter summarizes the outcomes of the project, its limitations, potential improvements and future research directions.

Chapter 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVANT LITERATURE

Phishing attacks are one of the most common cybersecurity threats, and they pose a significant risk to the security and integrity of online voting systems. Phishing attacks in voting systems can manipulate voters to disclose sensitive information or vote for a particular candidate or party. Visual cryptography is a promising technique for preventing phishing attacks in voting systems. These literature survey provides an overview of recent research on preventing phishing attacks on voting systems using visual cryptography -

1. This paper [1] proposes a novel fuzzy random grid-based approach where instead of converting images to binary, the method directly encrypts gray and color images using fuzzy random grids with decimal values between 0.0 and 1.0.

2. In the proposed paper[2], the voting and result phases will be run simultaneously, allowing the proposed system to realize the main notion of openly showing live results to each and every voter. A blockchain is a distributed ledger that duplicates and distributes transactions across the network of computers participating in the blockchain.

3. The proposed hybrid approach in this paper [3] combines the strengths of both data hiding and VC to achieve secure data hiding and extraction. To make the scheme more practical, an improved approach is presented for reducing the pixel expansion, increasing the visual quality of recovered image, and enhancing the data hiding efficiency.

4. The paper [4] introduces a novel decentralized lattice-based method for visual symmetric cryptography, dubbed Pico crypt that do image encryption and decryption to be more efficient. The decentralized lattice-based technique consists of two layers for improving the security level; the first layer is structured by

computational and algebraic functions, and the second layer is a simple but effective step based on American Standard Code for Information Interchange (ASCII) encoding.

5. This paper [6] proposes PVC scheme that utilizes simple modular arithmetic operations to achieve progressive image sharing. This approach can be impractical for large or sensitive images.

6. The system inn this paper [8] assures confidentiality as it uses homomorphic properties for calculation of votes in their encrypted form and decrypt to get only total votes though keeping it secure. Homomorphic encryption is the conversion of data into ciphertext that can be analyzed and worked with as if it were still in its original form. Homomorphic encryption enables complex mathematical operations to be performed on encrypted data without compromising the encryption.

2.2 TABLUAR FORM OF THE RELEVANT LITERATURE

S. No.	Paper Title	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results	Limitations
1.	"Visual Secret Sharing of Gray and Color Image using Fuzzy Random Grids"	Applied soft computing (2023)	C++/Python, Fuzzy logic libraries	Degree of Fuzziness, Robustness, visual decryption	Performance overhead, limited usability and capacity
2.	"Transparent Voting system using Blockchain"	Journal of the International Measurement Confederation (2023)	Blockchain and ReactJS	the voting and result phases will be run simultaneously, allowing the proposed system to realize the main notion of openly showing live results to each and every voter.	They are showing live results to the voters as well which can affect the voter's choice to whom they are voting.

3.	“A hybrid approach combining data hiding with visual cryptography for secure extraction of data hiding”	Journal of Information Security and Applications (2023)	steganography , Dividing the image	Security vis collab, Data Concealment	Performance overhead
4.	“Decentralized lattice-based method for visual symmetric cryptography”	Franklin open (2023)	C++/Python, Lattice cryptography libraries like Lizard library	Post-Quantum Security, Scalability, Decentralization	Computationally expensive, Limited adoption, key management is complex
5.	“progressive visual cryptography using simple modular arithmetic operations”	Journal of visual info and image communication (2021)	C++, Hashing, Cryptographic algorithms	Partial visualization, progressive decryption	Limited security, key management, limited flexibility.

6.	“Online voting system using Homomorphic encryption”	ITM web of conferences (2020)	Homomorphic encryption algorithm and Paillier cryptosystem	This system assures confidentiality as it uses homomorphic properties for calculation of votes in their encrypted form and decrypt to get only total votes though keeping it secure.	As the system uses homomorphic encryption technique, the system works to slow because this technique need a very high computational time.

Table Error! No text of specified style in document.**1 Literature Review**

2.3 KEY GAPS IN LITERATURE

All the proposed research papers performed very well but there were some key gaps in each of the paper like there were research papers which have Performance overhead, limited usability and capacity. Other than that, there was a paper that was showing live results to the voters as well which can affect the voter’s choice to whom they are voting. There was also a research paper which was Computationally expensive, have Limited adoption and key management was also complex. There was also a paper which used biometric censor for the authentication purpose but biometric censor is a thing that normally people don’t have at their home.

There were research papers which were not focusing on key management and complexity and had Limited security, limited Flexibility, Storage requirements and were Expensive. There was a system which was using homomorphic encryption technique, and as the system uses homomorphic encryption technique, the system works to slow because this technique needs a very high computational time. In the AES and DES comparison, research paper only compared AES and DES algorithm and not any other as other algorithms may have more avalanche effect. There was a comparative analysis that was unable to give a proper result and can’t able to analyze which technique is best to implement online voting system.

In conclusion, visual cryptography is a promising technique for preventing phishing attacks in voting systems. The proposed approaches in the literature survey use visual cryptography to generate multiple shares of the user's identity image, and the shares are used to authenticate the voter's identity. To validate the proposed methodologies using real-world datasets and to address the scalability problems of visual cryptography-based approaches, more study is required.

Chapter 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

3.1.1 FUNCTIONAL REQUIREMENT

The following can be included in functional requirements –

1. **Visual cryptography algorithm** – The system must have a secured and dependable visual cryptography algorithm that can create different shares for each voter's identification which make it possible for them to be safely merged in order to get the original data.

2. **Share distribution** – In order to prevent hacker's attacks, the system must be able to share different shares with the users only through different modes of communication like text message or emails.

3. **Share validation** – In order to make sure that each share that the system receives from the voter is correct and not been altered, each share must be validated.

4. **Share combining** – The system must provide a straightforward and understandable user interface that allows voters to combine their shares.

5. **Voter authentication** – In order to prevent any unwilling access in the voting process, the system must have a mechanism so that it can verify each voter's identity and authenticate the voter correctly.

6. **Result verification** – In order to guarantee the accuracy and fairness of the election, the system must permit independent verification of the voting results.

3.1.2 NON-FUNCTION REQUIREMENT

The following could be included in the non-functional requirements –

1. **Security** – The system must guarantee the privacy, confidentiality, and integrity of the data that will be stored and along with it guard the process from any kind of unauthorized access.

2. **Scalability** – The system must be scalable and capable of handling numerous voters at once and ready for future increases in the number of voters and elections.

3. **Usability** – The system must be simple to operate and use for all kind of users as there could be many users that don’t even know how to operate a site.

4. **Accessibility** – In order to guarantee all users to access and use the system, it must follow the accessibility standards and guidelines.

5. **Performance** –The system must be fast and efficient in order to avoid and stop any kind of delay or bottlenecks.

6. **Reliability** – The system must be reliable and dependable with little or no interference of any outer party.

Applications Interface	Languages
Frontend	ReactJS
Backend	Django
Database	PSQL
Terminal	Hyper & Command Prompt
App	VS Code
Server	AWS

Table 3 Error! No text of specified style in document.**1 Knowledge requirement**

3.2 PROJECT DESIGN AND ARCHITECTURE

3.2.1 PROJECT DESIGN

The Project had designed in such a way that it can be easier for all kind of users. The given diagram will show the flow chart of the project. In this system, firstly the admin would register and then after login, he would create elections. After setting the dates for election, he needs to add candidates and the voters. Once the candidates and voter’s details are entered by the admin, an email would be sent to all the voters, so that they can login through that mail and get into the voting process.

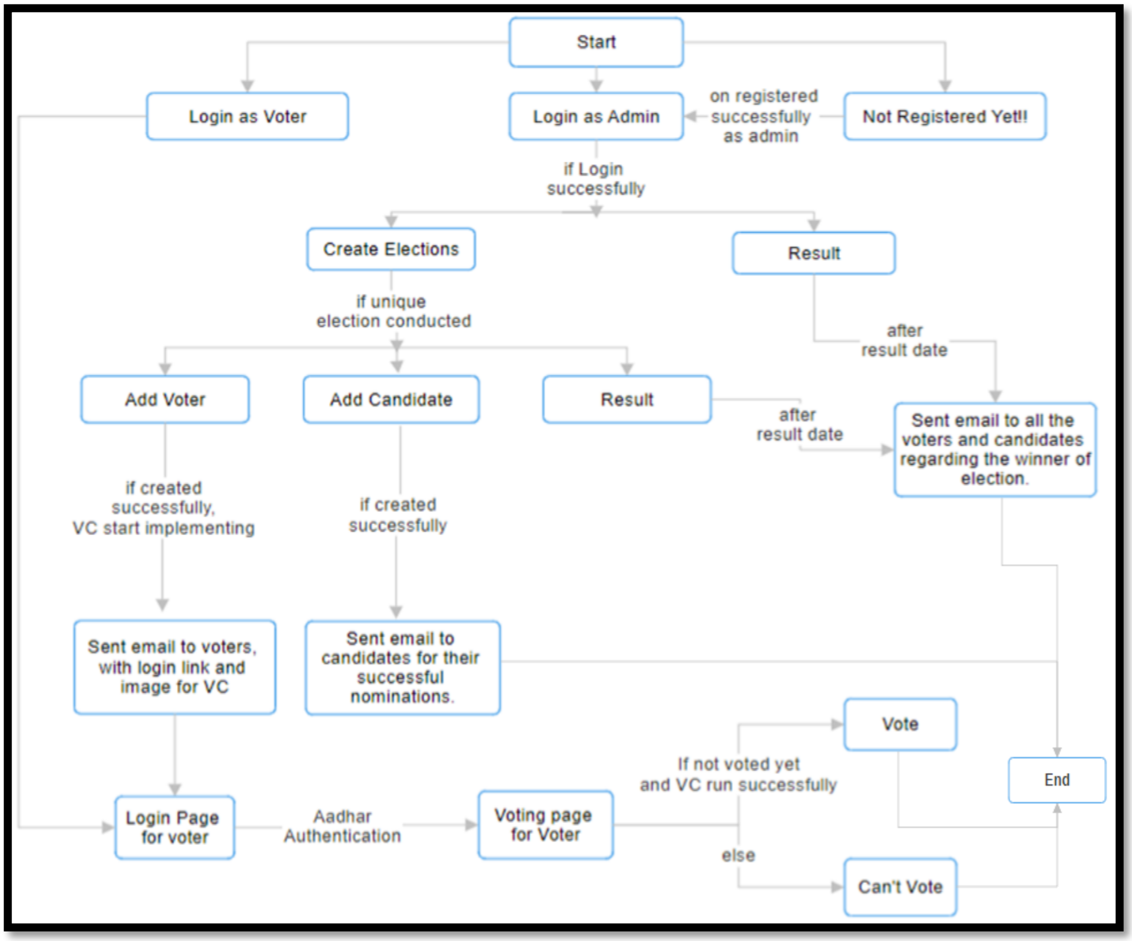

Start
Login as Voter
Login as Admin
on registered successfully as admin
Not Registered Yet!!
if Login successfully
Create Elections
Result
if unique election conducted
after result date
Add Voter
Add Candidate
Result
after result date
Sent email to all the voters and candidates regarding the winner of election.
if created successfully, VC start implementing
if created successfully
Sent email to voters, with login link and image for VC
Sent email to candidates for their successful nominations.
Vote
If not voted yet and VC run successfully
End
Login Page for voter
Aadhar Authentication
Voting page for Voter
else
Can't Vote

Figure 3.1: Proposed architecture

After successful Login, voters can vote once to one of the candidates listed there and make their favorite candidate win the elections and after he clicked the vote button, the visual cryptography algo start implementing. Once we reach the end date, no one can vote anymore and the results would be shown to the admin in terms of bar graph, pie chart and tabular form.

3.2.2 DATA FLOW DIAGRAM

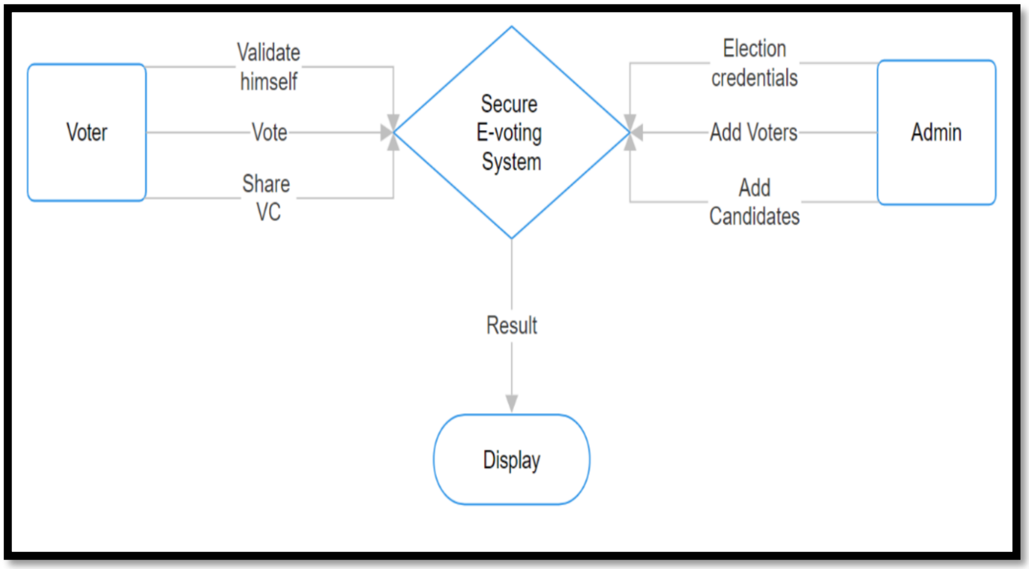


Figure 3.2: level0 DFD

3.2.3 ENTITY RELATION DIAGRAM

- ➔ Each election is associated with a user through the host foreign key.
- ➔ Each candidate is associated with a user through account holder foreign key.
- ➔ Each candidate is associated with multiple election instance through the election name (many to many relationship).
- ➔ Each voter is associated with a user through the host foreign key.
- ➔ Each voter is associated with multiple election instance through the election name (many to many relationship).
- ➔ The voted model seems to be tracking the relationship between voters and candidates.

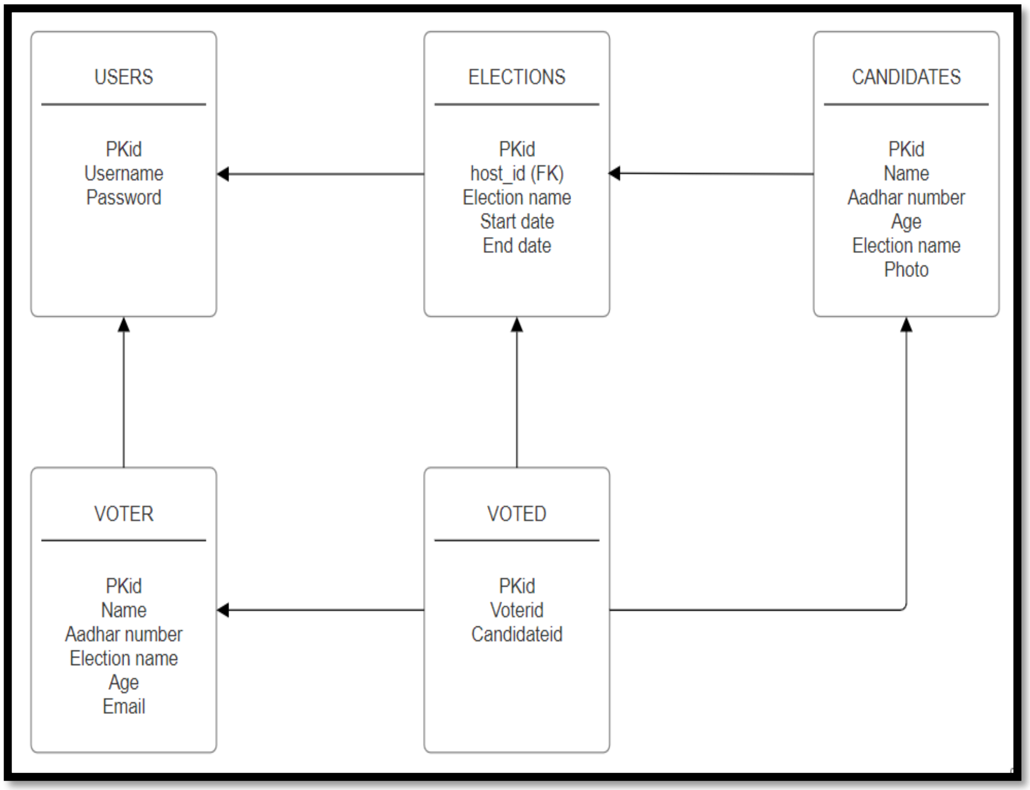


Figure 3.3: ER Diagram

3.3 IMPLEMENTATION

3.3.1 IMPLEMENTATION DETAILS

This section describes the implementation details of Securing voting system using visual cryptography –

- Static features of front end of Securing voting system using visual cryptography are implemented using ReactJS [14].

- Dynamic features of front end of Securing voting system using visual cryptography are implemented using ReactJS [14].

- Backend is implemented using Python with Django [15] framework. Code for Steganography and Visual Cryptography are written in python.

- For this system AWS server and PSQL database are used.

- This website is supported by JavaScript supporting browsers like Chrome, Firefox etc.

- The terminals use for the implementations are Hyper Terminal and VS code.

- The platform used for writing the code is VS Code and Digital Ocean is used for deploying the Backend server on cloud.

3.3.2 VISUAL CRYPTOGRAPHY ALGORITHM

In Visual cryptography [13], we have a text that we hide behind an image and the image is further divided into 2 or more images at the encryption level. For decryption all the images got superimposed and the original text came out.

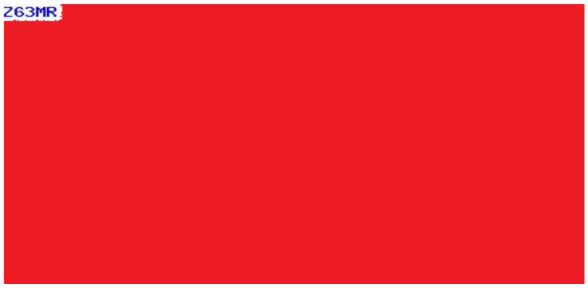


Figure 3.4a: Initial Captcha Image

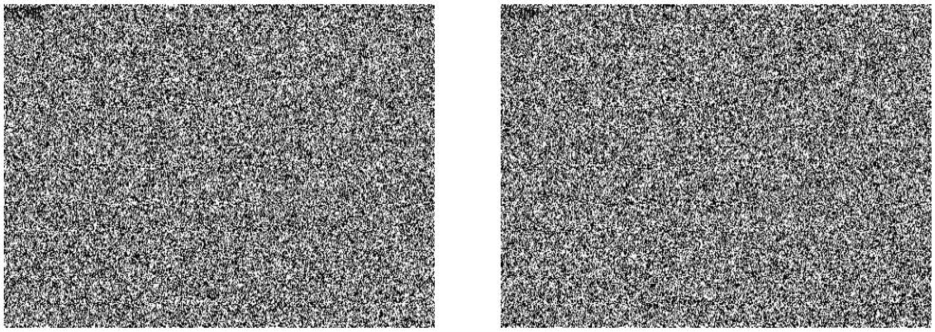


Figure 3.4b: Two shares in which the original captcha Image is divided

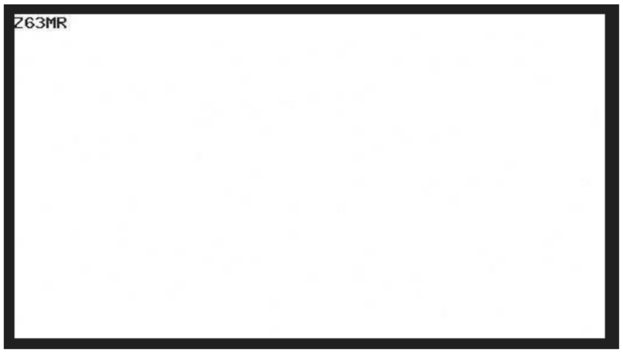


Figure 3.4c: Reconstructed captcha Image displayed on the page.

In this report we would typically describe how this technique works in simple terms, focusing on concepts like –

- ➔ **Encryption –** Explain how the original image or text is divided into shares using mathematical algorithms. Each share looks like random noise or patterns and doesn't give any clue about the original content.

- ➔ **Decryption –** Describe how combining these shares using a specific method, such as stacking or superimposing them, reveals the hidden message or image. Emphasize that this process doesn't involve complex computations or decoding but rather a simple visual merging.

- ➔ **Security** – Highlight the security aspect by mentioning that even if someone obtains one or more shares, they cannot decipher the original content without having all the shares and using the correct decryption method. This ensures confidentiality and privacy.

- ➔ **Applications –** Mention real-world applications of visual cryptography, such as secure image sharing, password recovery mechanisms, or watermarking techniques used to protect digital content.

Overall, the goal is to convey the idea that visual cryptography provides a simple yet effective way to secure and share information without the need for complex cryptographic techniques. It's a visually intuitive method that anyone can understand, making it versatile and applicable in various scenarios where data security is crucial.

3.3.3 MULTI FACTOR AUTHENTICATION

Other than the visual cryptography algorithm of authentication, we have also tried to add another two steps of authentications including OTP verification and email verification.

OTP Verification – The process of creating a special password or code for a one-time use is known as OTP verification. To put it simply, it's similar to a password that you create, use, and then forget. Usually, an app on your phone or a text message will generate the code and send it to your phone.

1. **Enhanced Security –** On top of your standard password, an OTP provides an additional layer of protection. Without the OTP code, someone can't log in even if they know your password.
2. **Prevents Unauthorized Access –** Even if hackers or other unauthorized users manage to get their hands on your password, they won't be able to access your account because the OTP is transient and changes every time.
3. **Lowers Risk of Account Hijacking –** By requiring both your password and the OTP code, OTP verification makes it considerably more difficult for someone to steal your account.

Email Verification – The process of email verification verifies that the email address a user provided when creating an account or registering is legitimate and actually belongs to them. In most cases, the user's email will receive a verification link or code that they must click or enter in order to verify their email address.

1. **Verifying the Identity of the User –** Email verification serves to confirm that the individual opening the account is the legitimate owner of the email address supplied. It stops other people from using fictitious or unapproved email addresses.
2. **Preventing Fraud and Spam –** Organizations can lessen the possibility of fraudulent activity, phony registrations, and spam accounts by verifying email addresses.

3. **Strengthening Security –** By verifying the authenticity of user accounts, email verification strengthens security even further. It stops illegal users from gaining access to private data or from carrying out tasks that call for account verification.

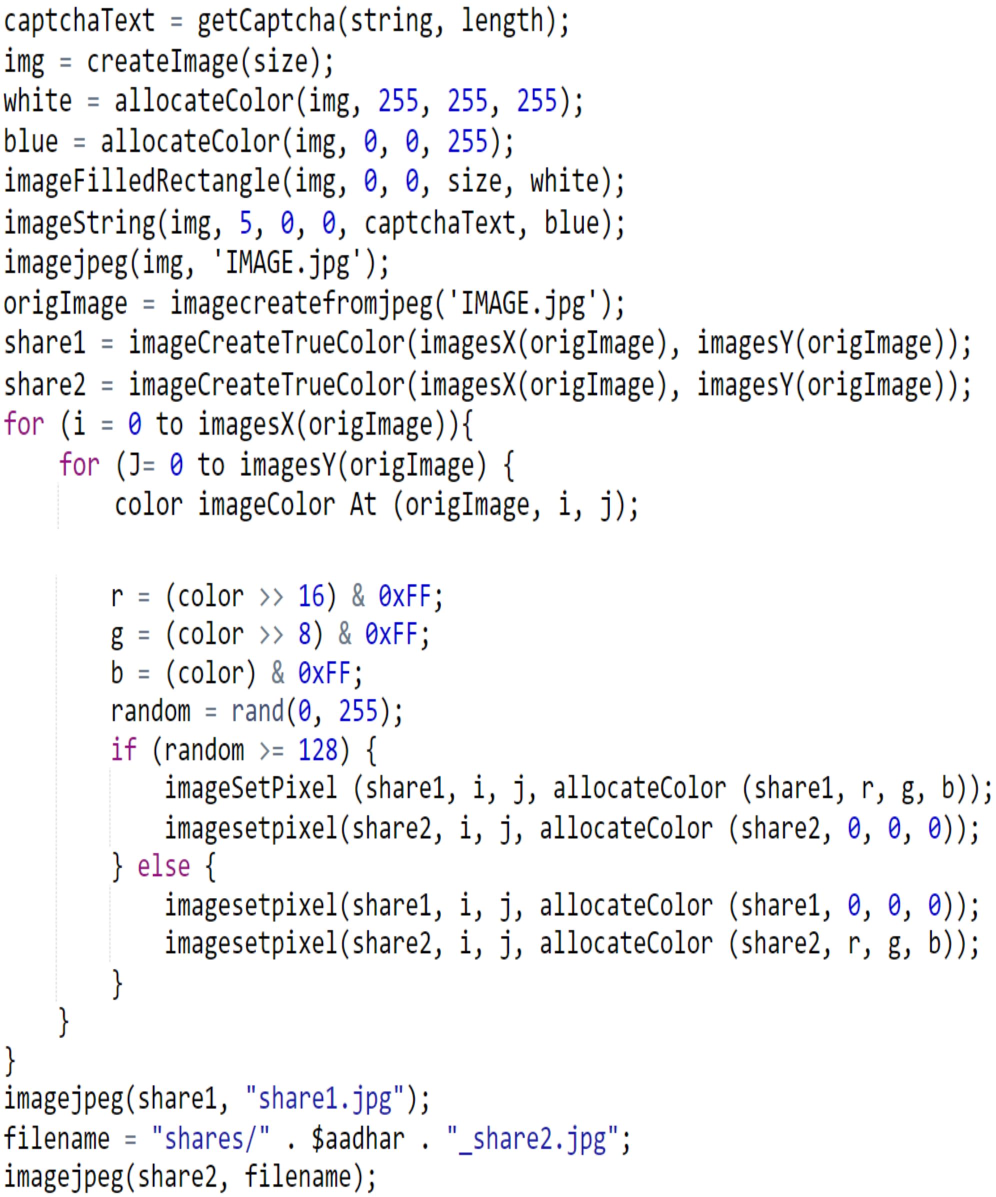
Improving the confidentiality, integrity and availability of user account –

1. Email verification: By verifying that the email address linked to the account is actually the owner of the account, email verification helps maintain account confidentiality. This guarantees that communications and sensitive information are sent only to email addresses that have been verified. Secrecy is improved by OTP verification, which makes sure that only the intended user—who has the code sent to their device.
2. Integrity: Email verification makes user accounts more secure by lessening the possibility of impersonation or account hijacking. It lowers the possibility of unauthorized changes or modifications to the account by confirming that the person creating it has access to the email address supplied. By lowering the possibility of unauthorized additions or modifications, the usage of OTP verification contributes to the preservation of user account integrity.
3. Availability: Email verification increases the number of steps involved in setting up an account, but in the end, it makes user accounts more accessible by safeguarding them against fraudulent or unauthorized access attempts. This makes it easier to guarantee that users can dependable and securely access their accounts. Although requiring an additional step during the login process, OTP verification ultimately helps to ensure that user accounts are available by protecting them from potential threats like account hijacking and unauthorized access attempts.

3.3.4 PSEUDO CODE FOR VISUAL CRYPTOGRAPHY

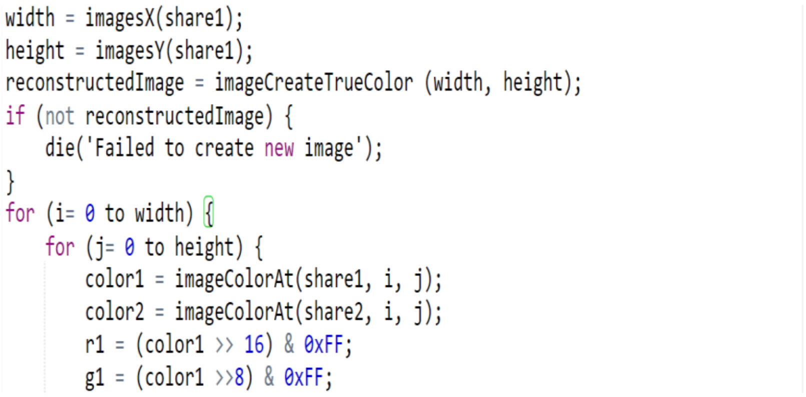
The pseudo code or algorithm for the visual cryptography looks like–

ENCRYPTION

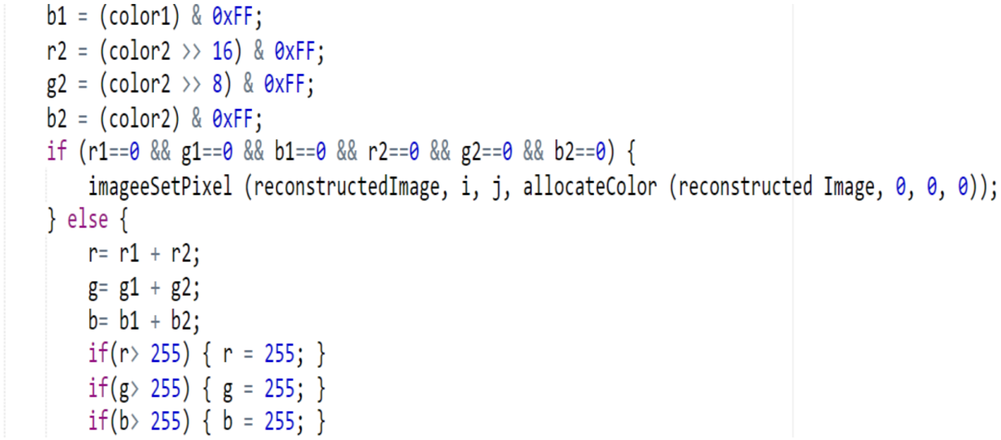


```
captchaText = getCaptcha(string, length);
img = createImage(size);
white = allocateColor(img, 255, 255, 255);
blue = allocateColor(img, 0, 0, 255);
imageFilledRectangle(img, 0, 0, size, white);
imageString(img, 5, 0, 0, captchaText, blue);
imagejpeg(img, 'IMAGE.jpg');
origImage = imagecreatefromjpeg('IMAGE.jpg');
share1 = imageCreateTrueColor(imagesX(origImage), imagesY(origImage));
share2 = imageCreateTrueColor(imagesX(origImage), imagesY(origImage));
for (i = 0 to imagesX(origImage)){
    for (J= 0 to imagesY(origImage) {
        color imageColor At (origImage, i, j);

        r = (color >> 16) & 0xFF;
        g = (color >> 8) & 0xFF;
        b = (color) & 0xFF;
        random = rand(0, 255);
        if (random >= 128) {
            imageSetPixel (share1, i, j, allocateColor (share1, r, g, b));
            imagesetpixel(share2, i, j, allocateColor (share2, 0, 0, 0));
        } else {
            imagesetpixel(share1, i, j, allocateColor (share1, 0, 0, 0));
            imagesetpixel(share2, i, j, allocateColor (share2, r, g, b));
        }
    }
}
imagejpeg(share1, "share1.jpg");
filename = "shares/" . $aadhar . "_share2.jpg";
imagejpeg(share2, filename);
```

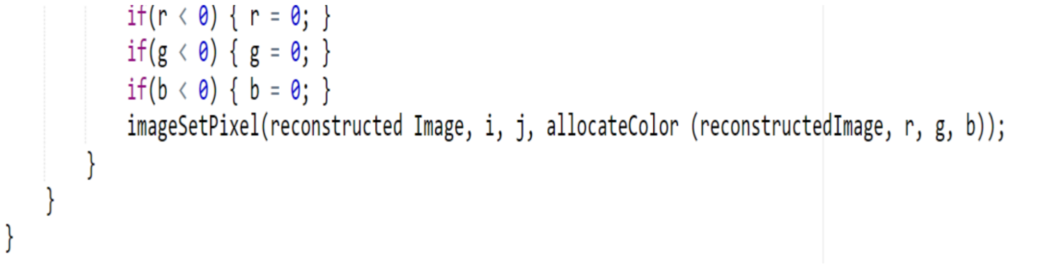
DECRYPTION



```
width = imagesX(share1);
height = imagesY(share1);
reconstructedImage = imageCreateTrueColor (width, height);
if (not reconstructedImage) {
    die('Failed to create new image');
}
for (i= 0 to width) {
    for (j= 0 to height) {
        color1 = imageColorAt(share1, i, j);
        color2 = imageColorAt(share2, i, j);
        r1 = (color1 >> 16) & 0xFF;
        g1 = (color1 >>8) & 0xFF;
```



```
        b1 = (color1) & 0xFF;
        r2 = (color2 >> 16) & 0xFF;
        g2 = (color2 >> 8) & 0xFF;
        b2 = (color2) & 0xFF;
        if (r1==0 && g1==0 && b1==0 && r2==0 && g2==0 && b2==0) {
            imageeSetPixel (reconstructedImage, i, j, allocateColor (reconstructed Image, 0, 0, 0));
        } else {
            r= r1 + r2;
            g= g1 + g2;
            b= b1 + b2;
            if(r> 255) { r = 255; }
            if(g> 255) { g = 255; }
            if(b> 255) { b = 255; }
```



```
            if(r < 0) { r = 0; }
            if(g < 0) { g = 0; }
            if(b < 0) { b = 0; }
            imageSetPixel(reconstructed Image, i, j, allocateColor (reconstructedImage, r, g, b));
        }
    }
}
```

3.3.5 VISUAL CRYPTOGRAPHY ENCRYPTION

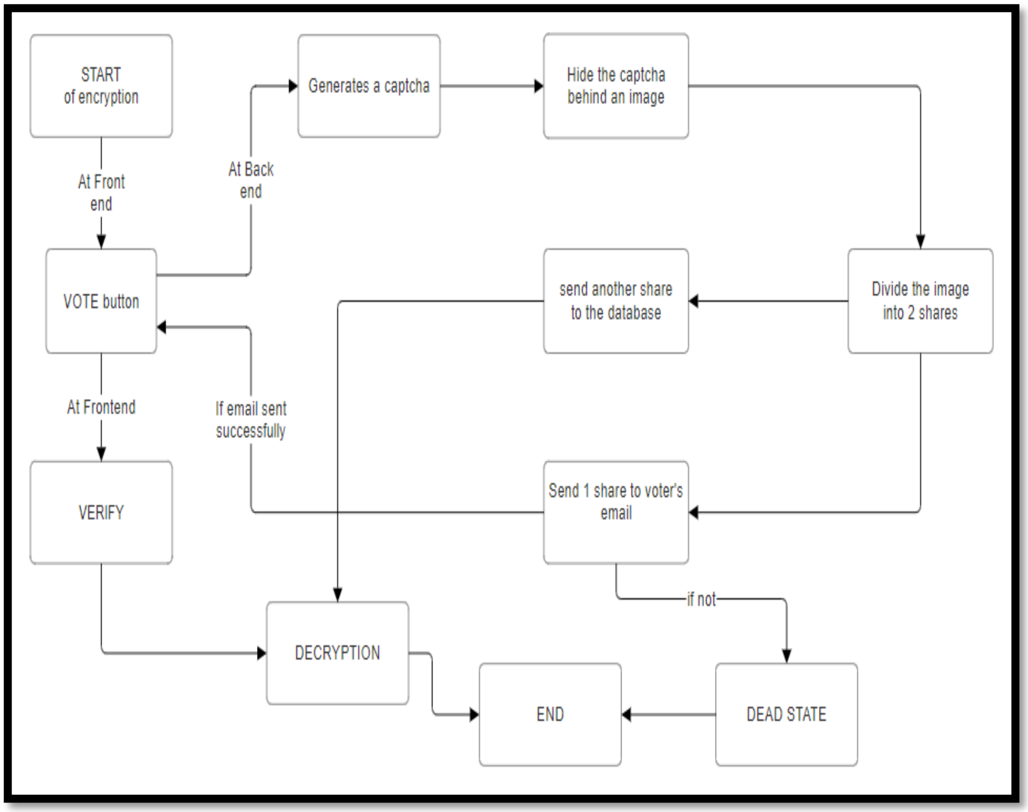

START of encryption
Generates a captcha
Hide the captcha behind an image
At Front end
At Back end
VOTE button
send another share to the database
Divide the image into 2 shares
At Frontend
If email sent successfully
VERIFY
Send 1 share to voter's email
if not
DECRYPTION
END
DEAD STATE

Figure 3.5: VC Encryption

At the time of encryption, firstly the voter gets the link for voting, where he will add his details and get an OTP on registered mobile number. After the completion of all these steps the encryption part of the visual cryptography got started. First the voter clicks on the VOTE button, backend generates a captcha, hide that captcha behind an image and then divide the image further into two shares. First share is saved in the database and the other share get sent to the voter on the registered email id.

If the things get into right direction than the voter will be directed to the verifying page and if not than he need to redo the tasks.

3.3.6 VISUAL CRYPTOGRAPHY DECRYPTION

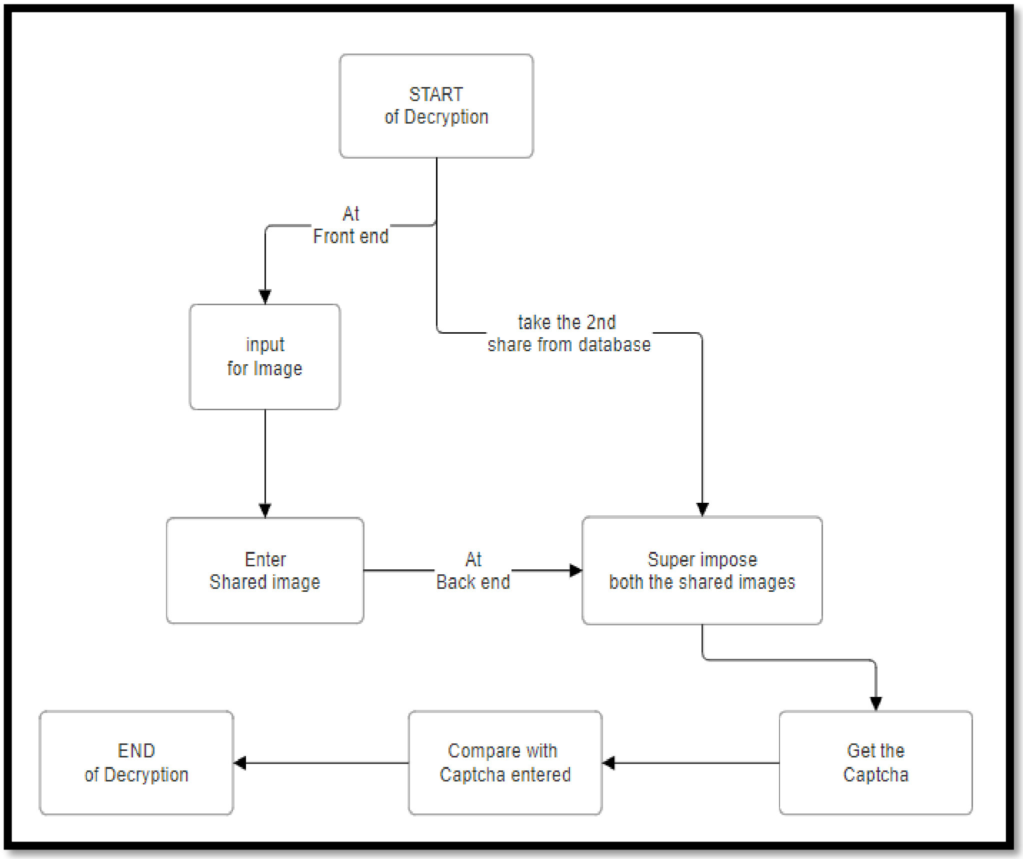
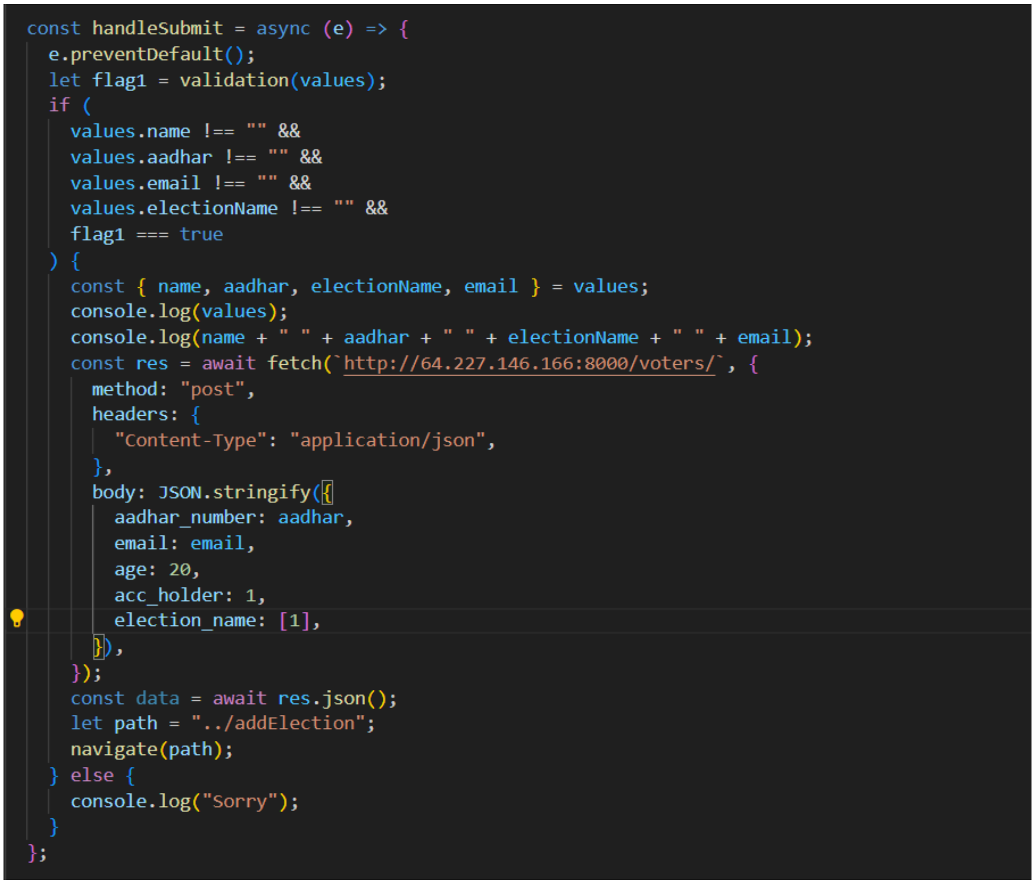


Figure 3.6: VC Decryption

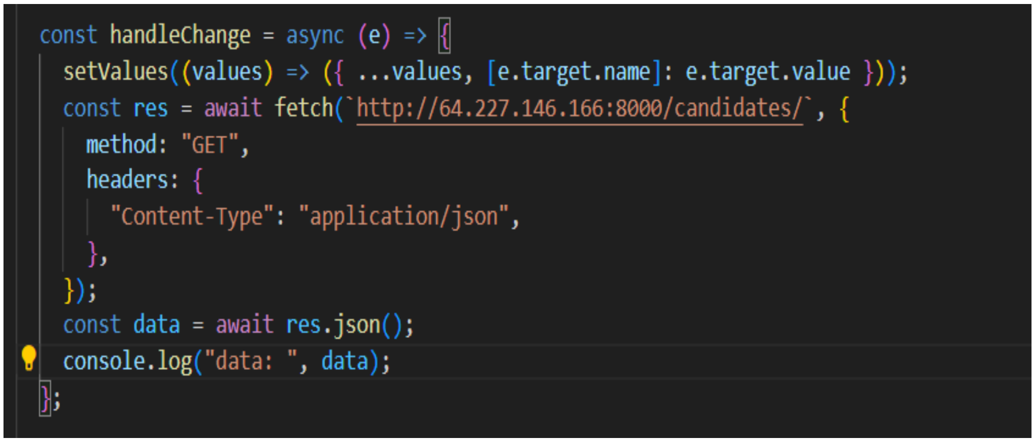
In the decryption part, firstly the voter needs to enter the image that he has received on his registered email. If the image is valid than it will continue with the process otherwise, he will receive the message of image or share not valid.
If the share is valid than the share or image will go to backend, each pixel of both the shares gets matched with each other and if both the shares are correct, the captcha get generated and user need to enter it. On entering the correct captcha, the vote has been successfully done and shown into the result section.

3.3.7 CODE SNIPPETS



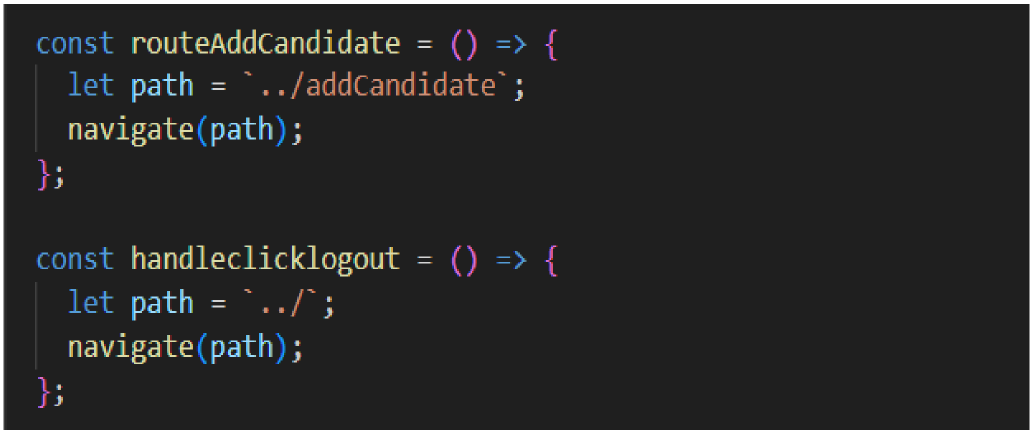
```
const handleSubmit = async (e) => {
  e.preventDefault();
  let flag1 = validation(values);
  if (
    values.name !== "" &&
    values.aadhar !== "" &&
    values.email !== "" &&
    values.electionName !== "" &&
    flag1 === true
  ) {
    const { name, aadhar, electionName, email } = values;
    console.log(values);
    console.log(name + " " + aadhar + " " + electionName + " " + email);
    const res = await fetch(`http://64.227.146.166:8000/voters/`, {
      method: "post",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({
        aadhar_number: aadhar,
        email: email,
        age: 20,
        acc_holder: 1,
        election_name: [1],
      }),
    });
    const data = await res.json();
    let path = "../addElection";
    navigate(path);
  } else {
    console.log("Sorry");
  }
};
```

Figure 3.7: Snippet for post request



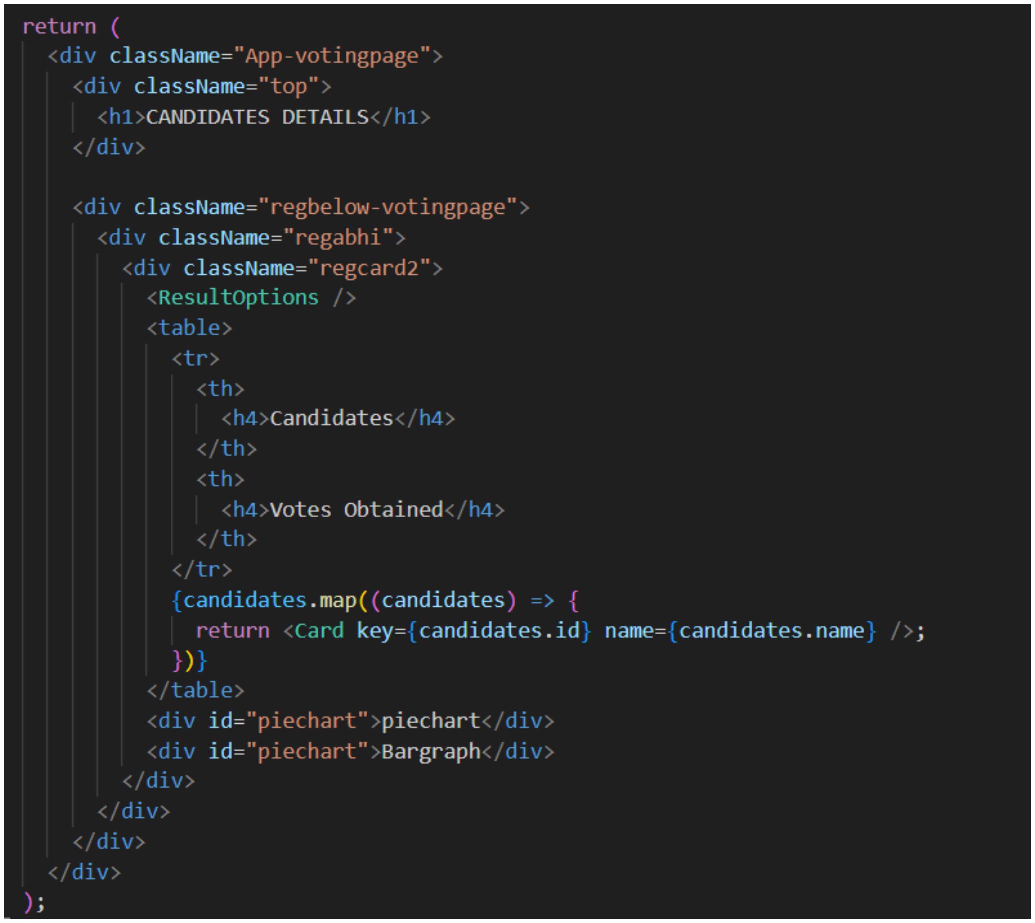
```
const handleChange = async (e) => {
  setValues((values) => ({ ...values, [e.target.name]: e.target.value }));
  const res = await fetch(`http://64.227.146.166:8000/candidates/`, {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  });
  const data = await res.json();
  console.log("data: ", data);
};
```

Figure 3.8: Snippet for get request



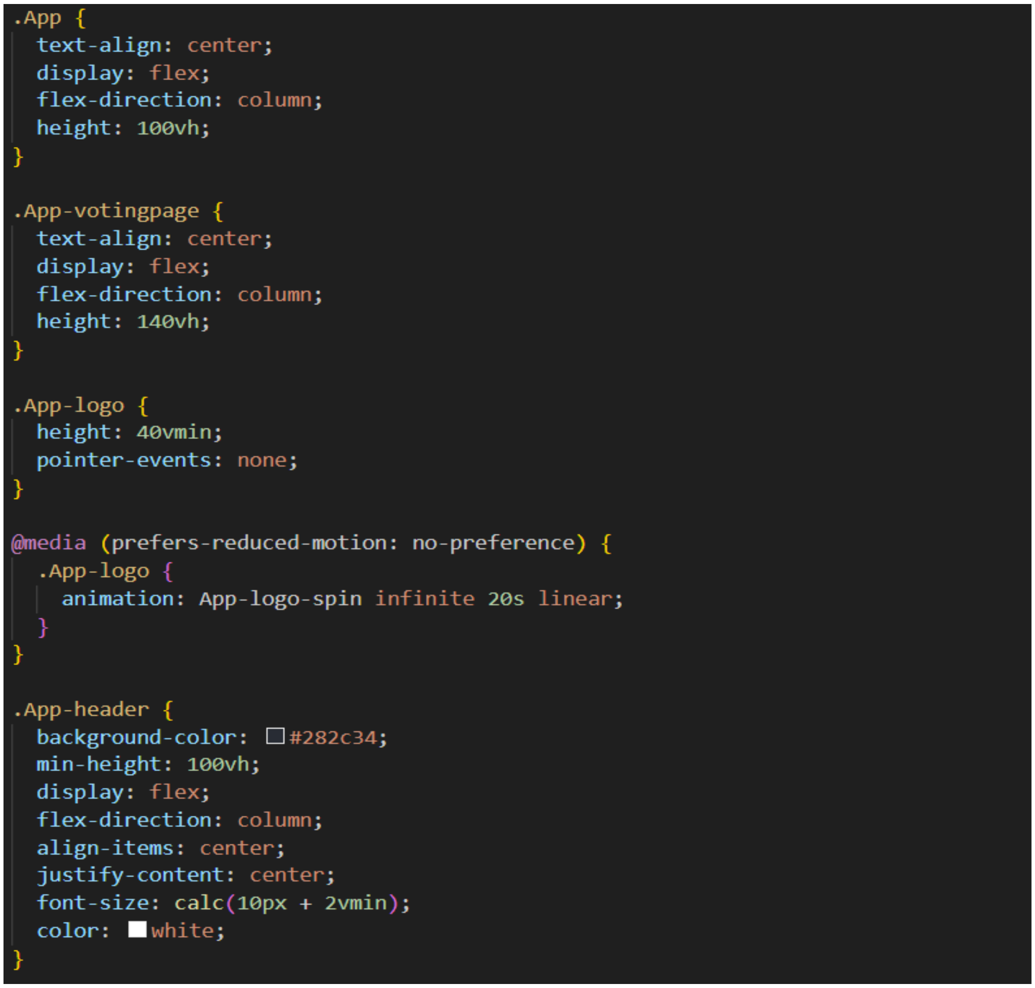
```
const routeAddCandidate = () => {
  let path = `../addCandidate`;
  navigate(path);
};

const handleclicklogout = () => {
  let path = `../`;
  navigate(path);
};
```

Figure 3.9: Snippet for navigation



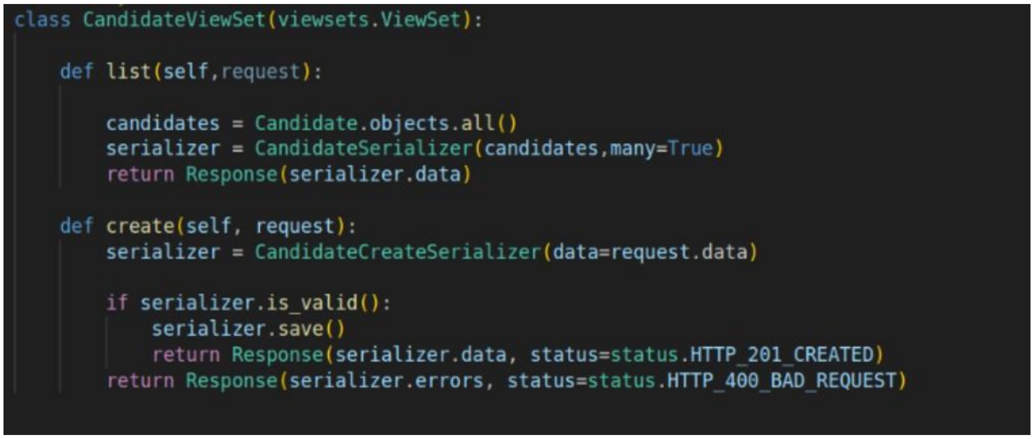
```
return (
  <div className="App-votingpage">
    <div className="top">
      <h1>CANDIDATES DETAILS</h1>
    </div>

    <div className="regbelow-votingpage">
      <div className="regabhi">
        <div className="regcard2">
          <ResultOptions />
          <table>
            <tr>
              <th>
                <h4>Candidates</h4>
              </th>
              <th>
                <h4>Votes Obtained</h4>
              </th>
            </tr>
            {candidates.map((candidates) => {
              return <Card key={candidates.id} name={candidates.name} />;
            })}
          </table>
          <div id="piechart">piechart</div>
          <div id="piechart">Bargraph</div>
        </div>
      </div>
    </div>
  </div>
);
```

Figure 3.10: Snippet for UI



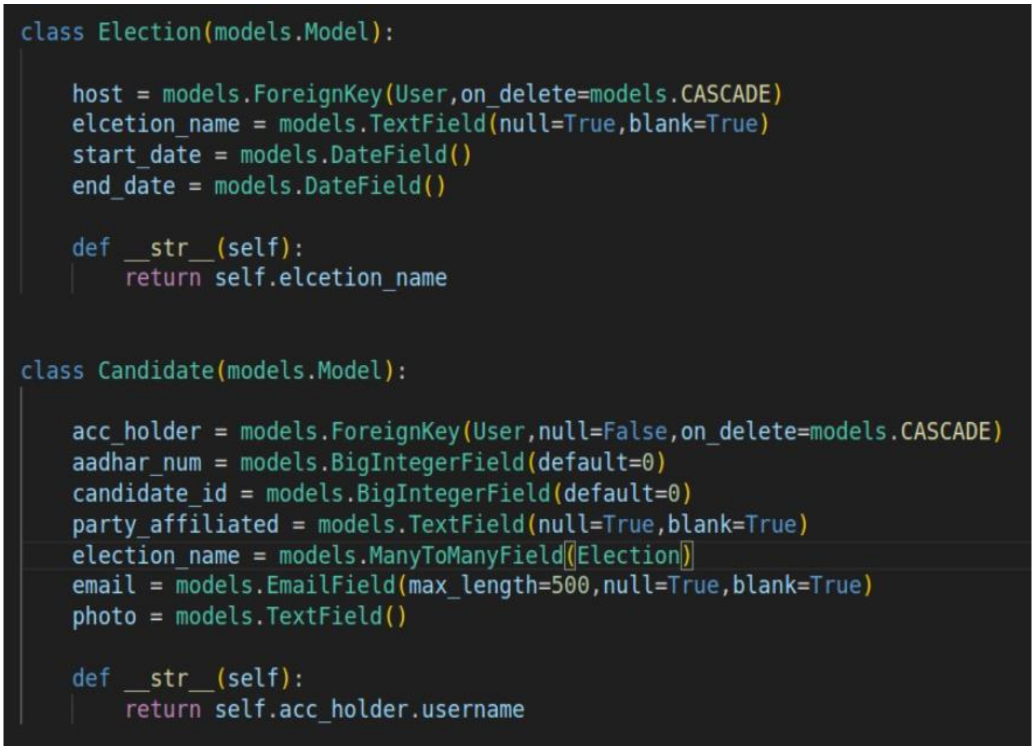
```
.App {
  text-align: center;
  display: flex;
  flex-direction: column;
  height: 100vh;
}

.App-votingpage {
  text-align: center;
  display: flex;
  flex-direction: column;
  height: 140vh;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: □#282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: ■white;
}
```

Figure 3.11: Snippet for Styling



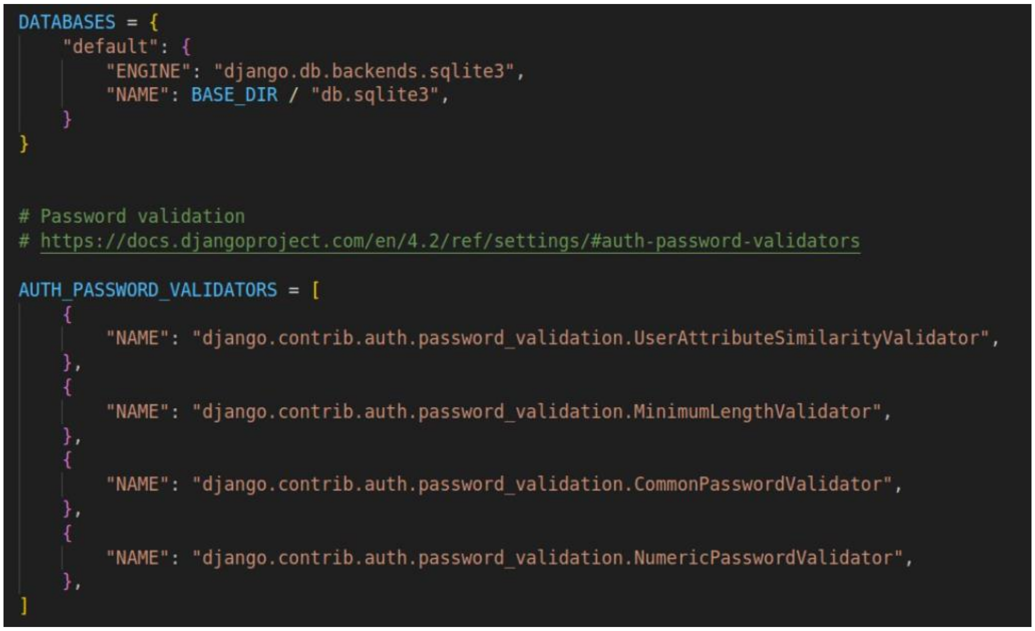
```
class CandidateViewSet(viewsets.ViewSet):

    def list(self,request):

        candidates = Candidate.objects.all()
        serializer = CandidateSerializer(candidates,many=True)
        return Response(serializer.data)

    def create(self, request):
        serializer = CandidateCreateSerializer(data=request.data)

        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Figure 3.12: Snippet for Get and Post request at backend



```python
class Election(models.Model):

    host = models.ForeignKey(User,on_delete=models.CASCADE)
    elcetion_name = models.TextField(null=True,blank=True)
    start_date = models.DateField()
    end_date = models.DateField()

    def __str__(self):
        return self.elcetion_name


class Candidate(models.Model):

    acc_holder = models.ForeignKey(User,null=False,on_delete=models.CASCADE)
    aadhar_num = models.BigIntegerField(default=0)
    candidate_id = models.BigIntegerField(default=0)
    party_affiliated = models.TextField(null=True,blank=True)
    election_name = models.ManyToManyField(Election)
    email = models.EmailField(max_length=500,null=True,blank=True)
    photo = models.TextField()

    def __str__(self):
        return self.acc_holder.username
```

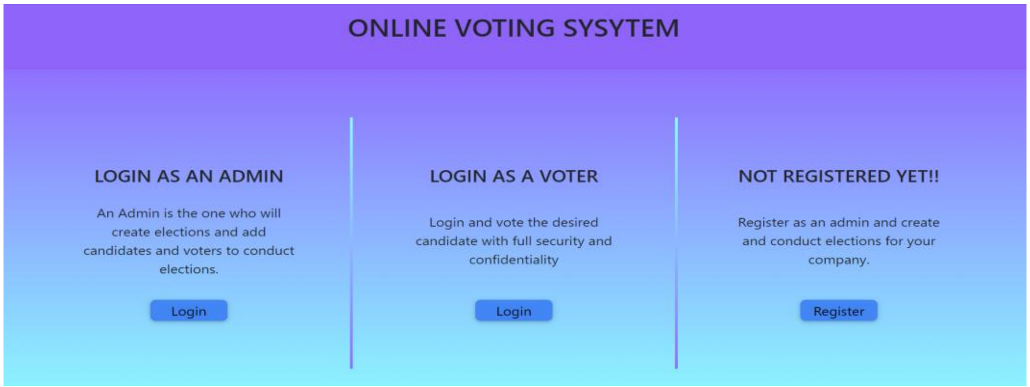
Figure 3.13: Snippet for object relational models



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


# Password validation
# https://docs.djangoproject.com/en/4.2/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",
    },
]
```

Figure 3.14: Snippet for configurations

3.3.8 SCREEN SHOTS OF VARIOUS STAGES OF PROJECT



ONLINE VOTING SYSYTEM

LOGIN AS AN ADMIN

An Admin is the one who will create elections and add candidates and voters to conduct elections.

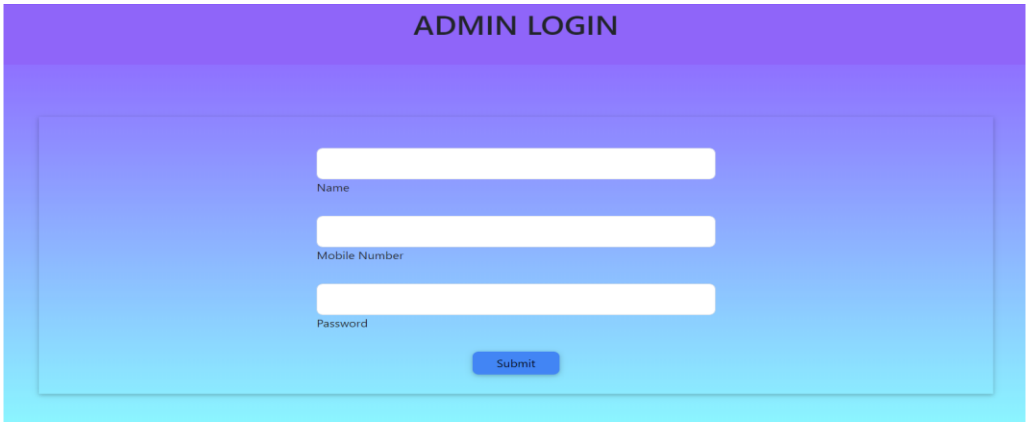
LOGIN AS A VOTER

Login and vote the desired candidate with full security and confidentiality

NOT REGISTERED YET!!

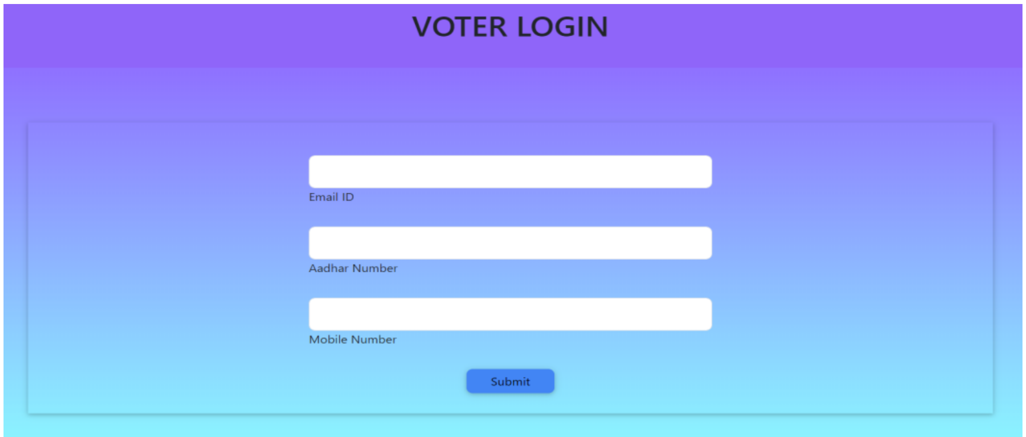
Register as an admin and create and conduct elections for your company.

Figure 3.15: Page to select either register or login



ADMIN LOGIN

Figure 3.16: Admin login Page



VOTER LOGIN

Figure 3.17: Voter login Page

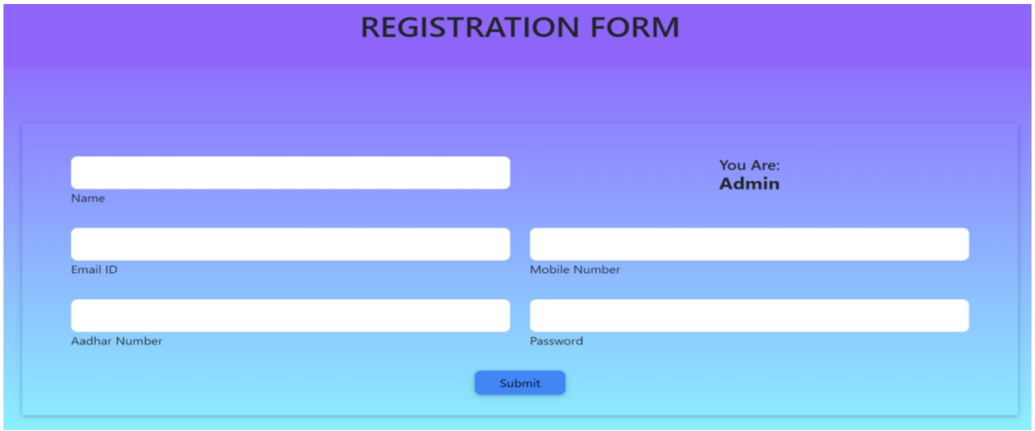
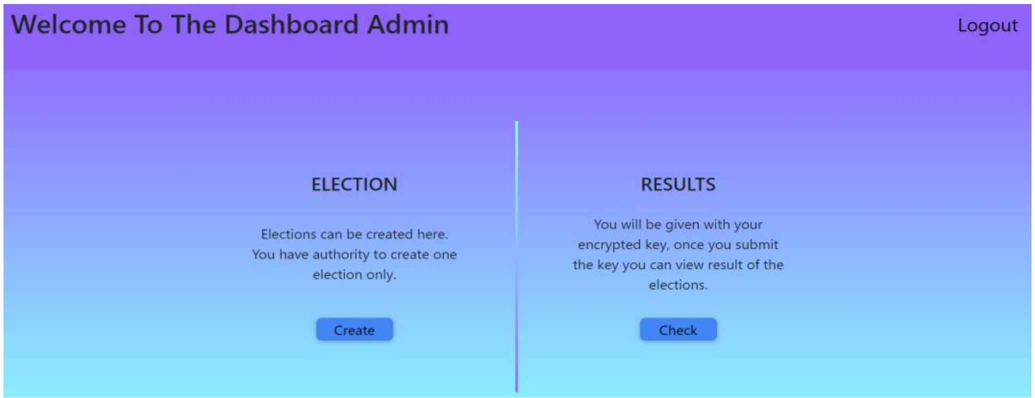


Figure 3.18: Admin Registration Page



Elections can be created here. You have authority to create one election only.

You will be given with your encrypted key, once you submit the key you can view result of the elections.

Figure 3.19: Admin Dashboard Page

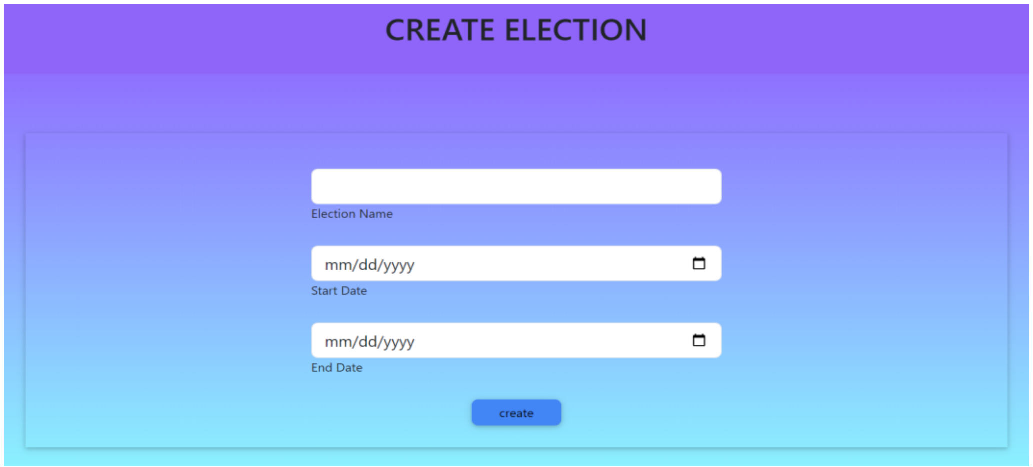
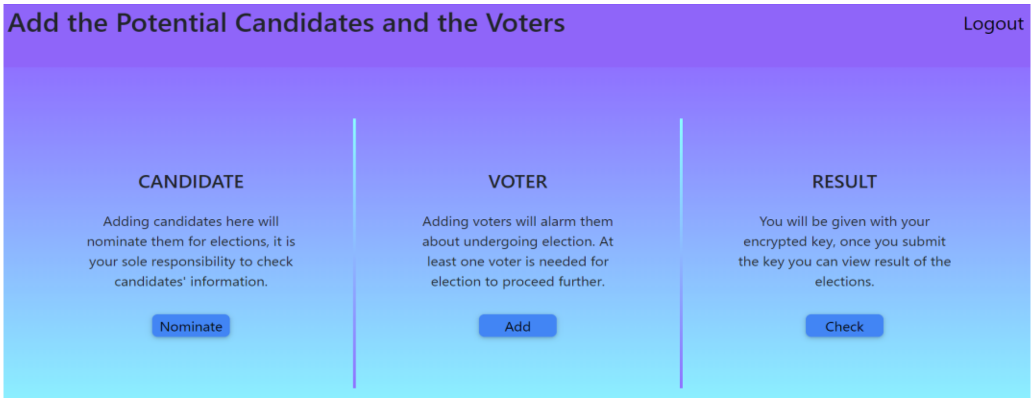


Figure 3.20: Create Election Page


Add the Potential Candidates and the Voters

CANDIDATE

Adding candidates here will nominate them for elections, it is your sole responsibility to check candidates' information.

VOTER

Adding voters will alarm them about undergoing election. At least one voter is needed for election to proceed further.

RESULT

You will be given with your encrypted key, once you submit the key you can view result of the elections.

Figure 3.21: Add Election Page

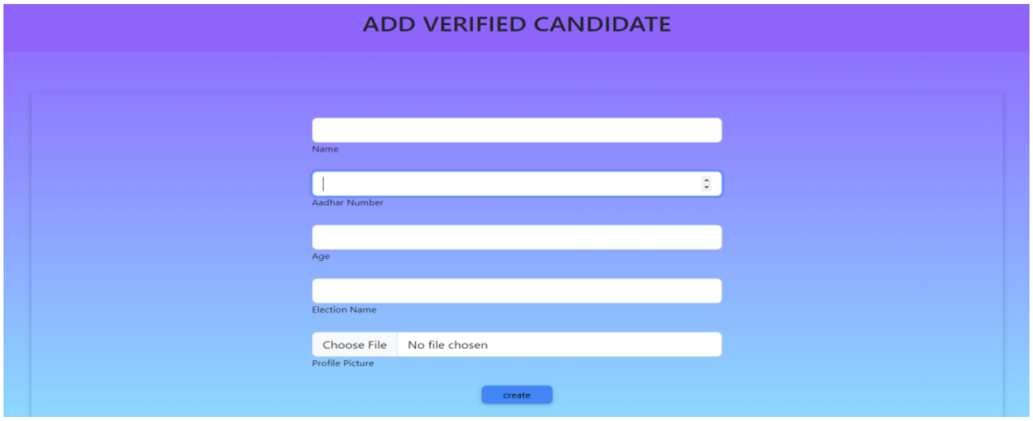

ADD VERIFIED CANDIDATE

Figure 3.22: Add Candidate Page

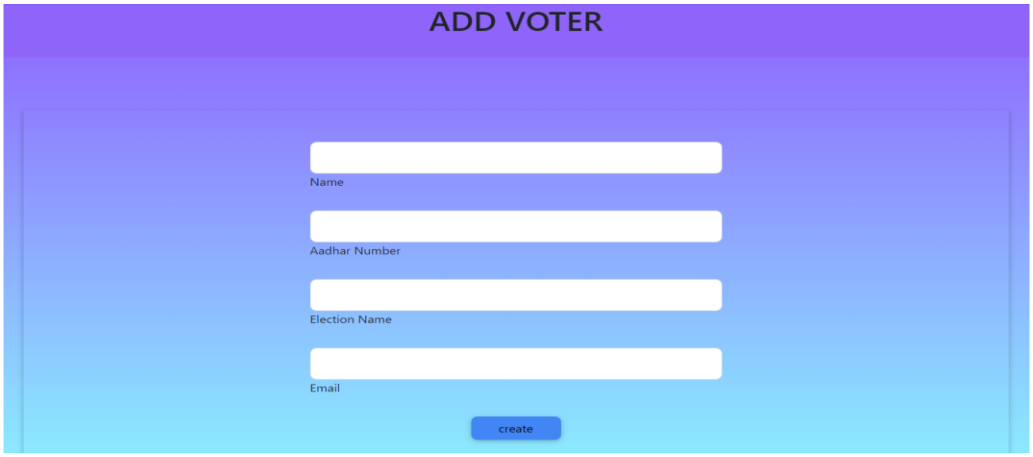
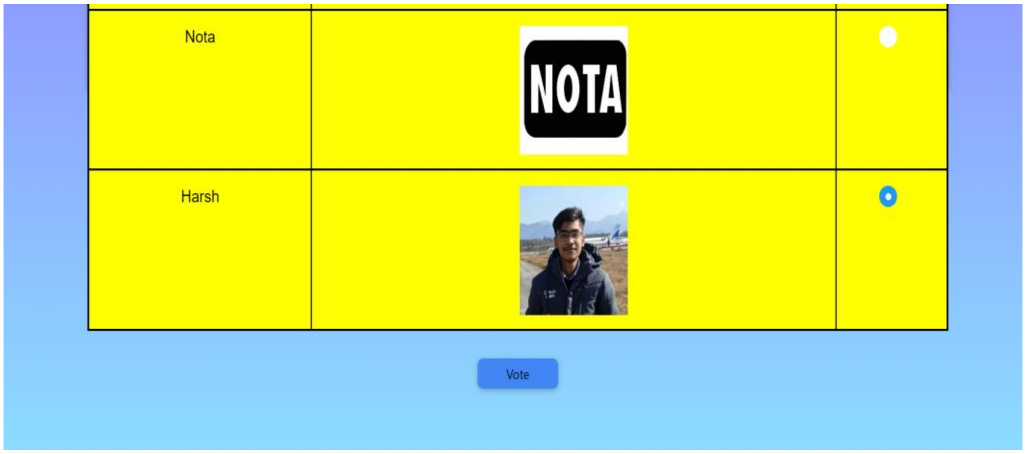
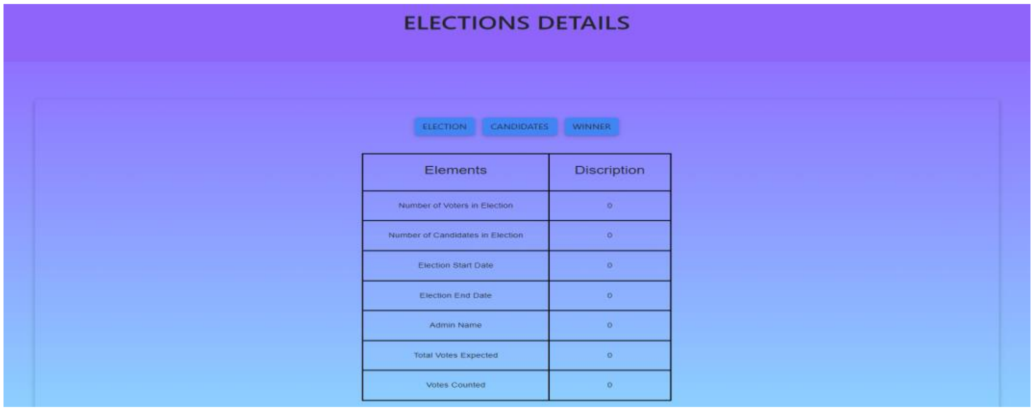

ADD VOTER

Figure 3.23: Add Voter Page



Nota		○
Harsh		●

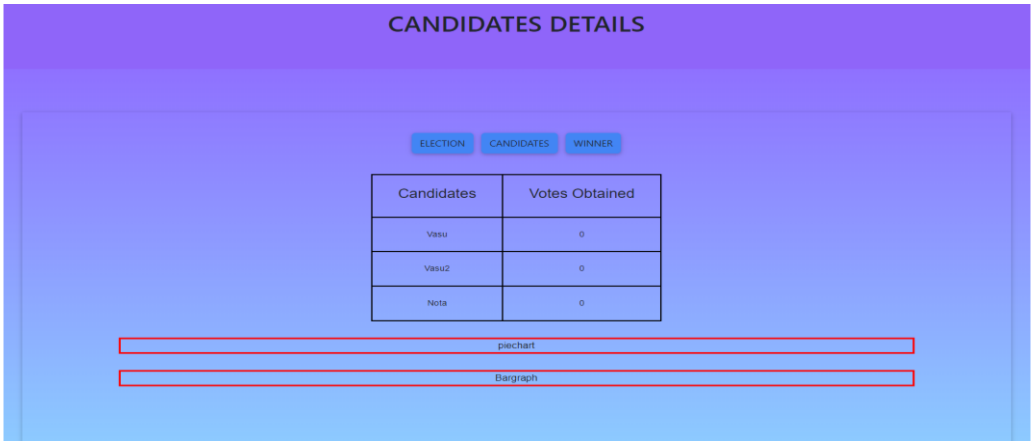
Figure 3.24: Voting Page



ELECTIONS DETAILS

Elements	Discription
Number of Voters in Election	0
Number of Candidates in Election	0
Election Start Date	0
Election End Date	0
Admin Name	0
Total Votes Expected	0
Votes Counted	0

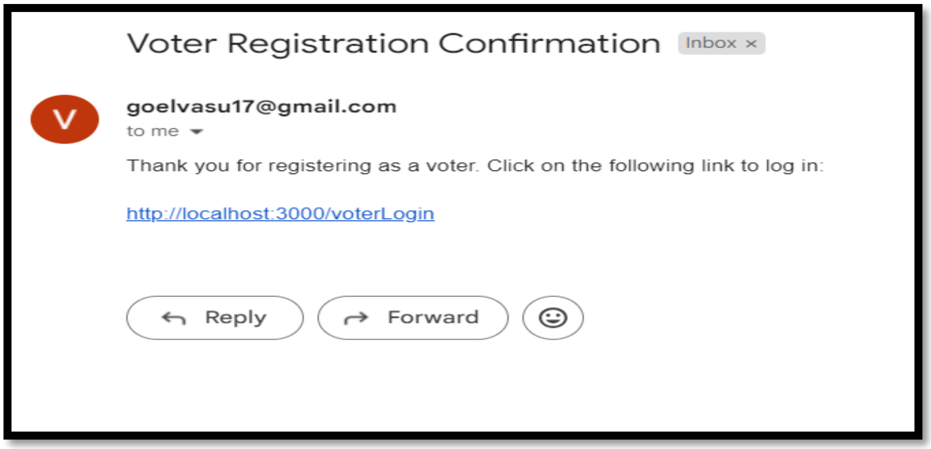
Figure 3.25: Election Details Page



CANDIDATES DETAILS

Candidates	Votes Obtained
Vasu	0
Vasu2	0
Nota	0

Figure 3.26: Candidate Details Page



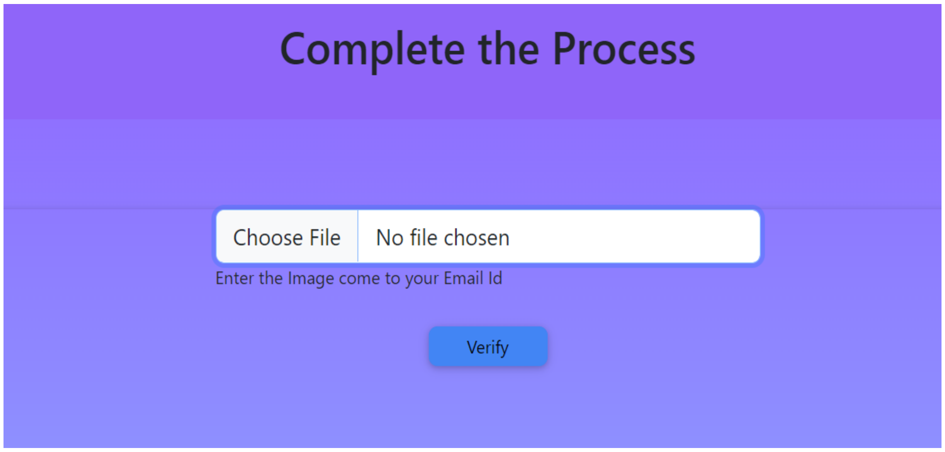
Voter Registration Confirmation

goelvasu17@gmail.com
to me

Thank you for registering as a voter. Click on the following link to log in:

http://localhost:3000/voterLogin

Figure 3.27: Voter receiving voting link

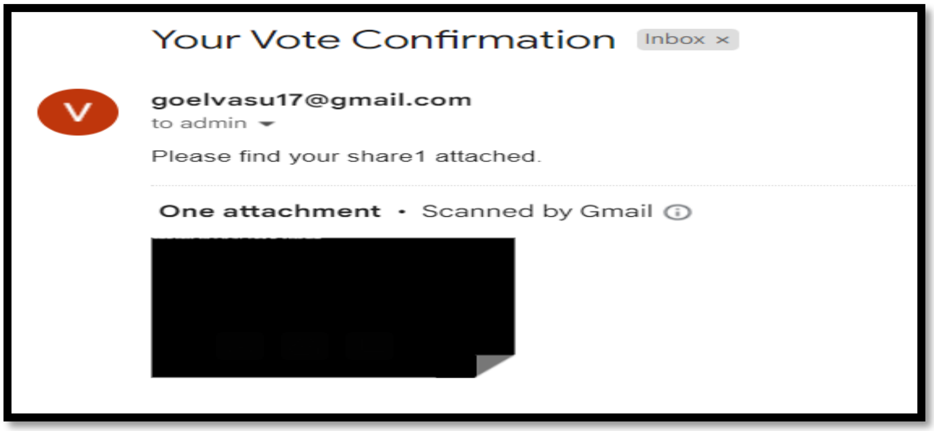


Complete the Process

Choose File | No file chosen

Enter the Image come to your Email Id

Figure 3.28: Visual Cryptography check



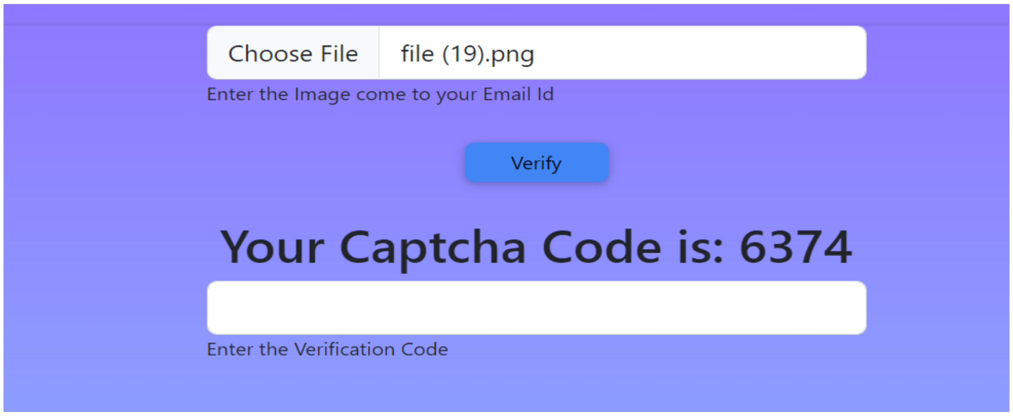
Your Vote Confirmation

goelvasu17@gmail.com
to admin

Please find your share1 attached.

One attachment • Scanned by Gmail

Figure 3.29: Voter receiving VC share



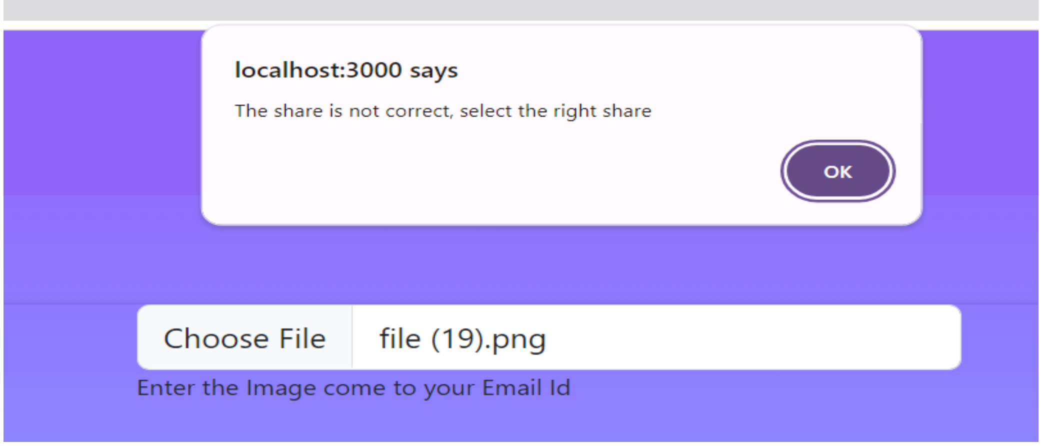
file (19).png

Enter the Image come to your Email Id

Your Captcha Code is: 6374

Enter the Verification Code

Figure 3.30: Successfully generated a captcha if share is correct



localhost:3000 says

The share is not correct, select the right share

file (19).png

Enter the Image come to your Email Id

Figure 3.31: Received an alert if share is not correct

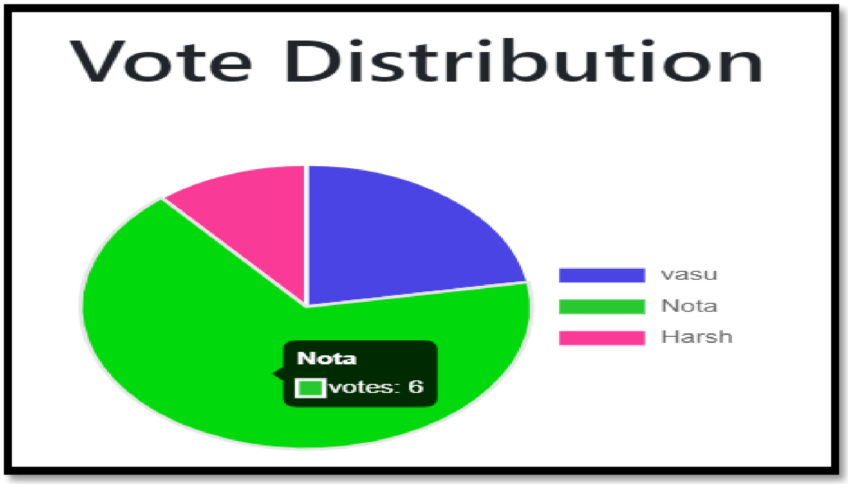
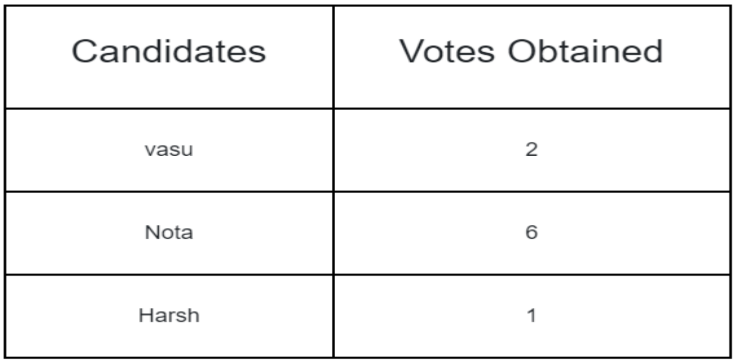


Figure 3.32: Result pie chart



Candidates	Votes Obtained
vasu	2
Nota	6
Harsh	1

Table 3 Error! No text of specified style in document.**2** Candidate Result

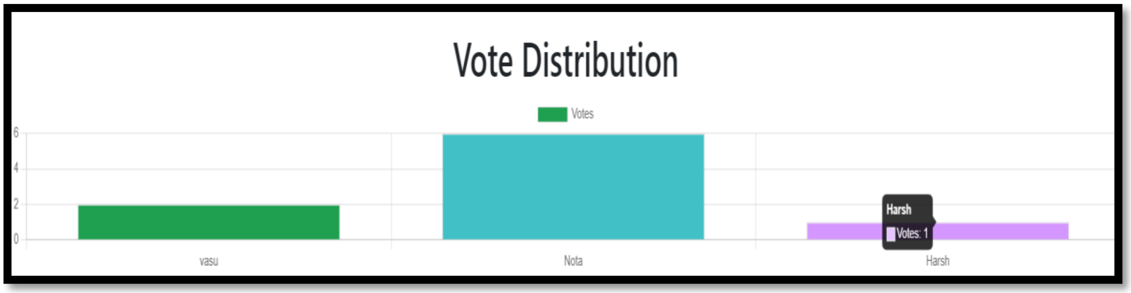
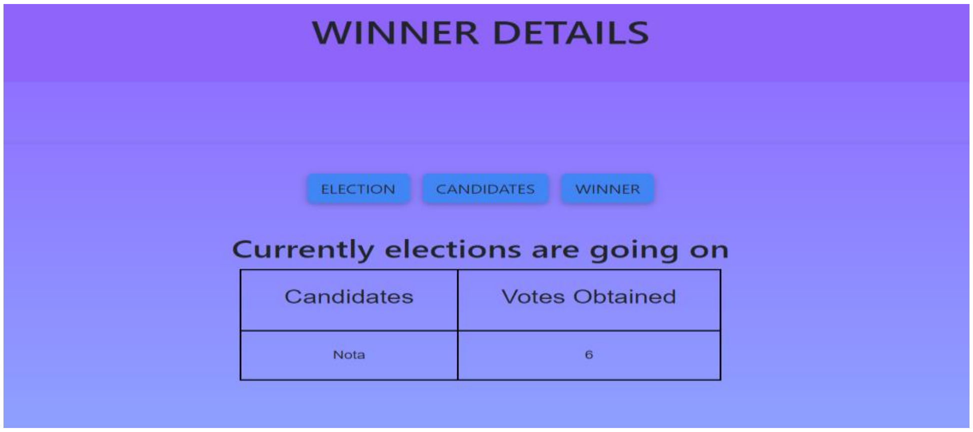


Figure 3.33: Result Bar Graph



WINNER DETAILS

Currently elections are going on

Candidates	Votes Obtained
Nota	6

Figure 3.34: Winner Page

3.4 KEY CHALLENGES

Different key challenges that we faced during the completion of our project were -

1. **Scope Creep** – Expanding the project beyond the original plan has increased costs and cause delays which lead the project to be delayed than expected.

2. **Resource Constraints** – Limited availability of skilled team members, budget restrictions, or insufficient time has impacted project progress but we did our best and tried to not make it an issue.

3. **Lack of Communication** – Ineffective communication within the team had caused some misunderstandings and hence let to delay in the project management.

4. **Technological Challenges** – As we were working in different systems and to connect both the systems, it become difficult for us as there was no unpaid software or cloud was available.

5. **Quality Assurance** – To ensure the quality of project, the time in project making automatically increase than the expected, so we tried not to compromise the quality within the given time.

To tackle these challenges, thorough project planning, regular reassessment, and effective communication were the key. It was important to collaborate within the team and maintain the flexibility in responding to any change or circumstances.

Chapter 4: TESTING

4.1 TESTING STRATEGY

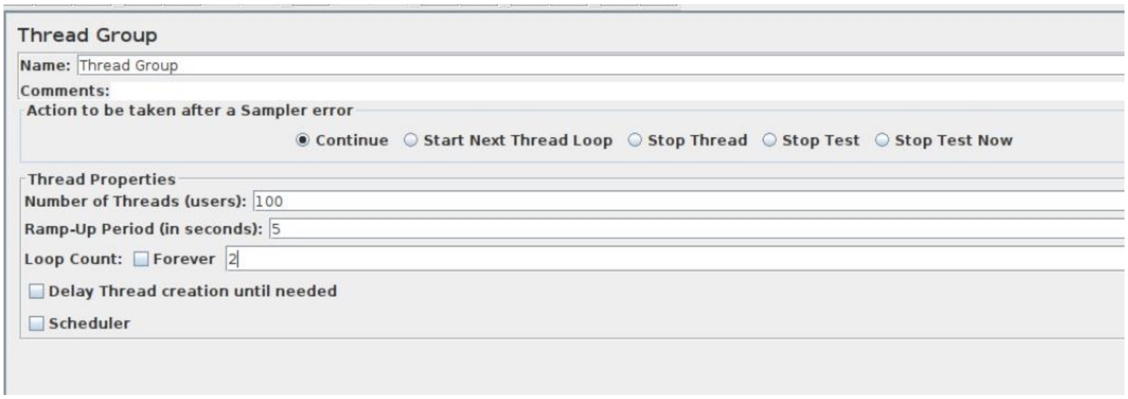
We use the Apache JMeter tool to test the performance of API’s and implement the best strategy. The main purpose of testing strategy is to check the system’s ability that it can identify the problems, detect if any thing not working so that we can solve them if any. Also, it helps to get the response time so that it can be compared with other systems present in the market.

To achieve this goal, we have assumed, created and executed some test cases that simulated user behaviour and operational scenarios. These factors include stress tests that measure the body's ability to withstand weight, endurance tests that measure long-term performance, and performance tests that measure the ability to perform more tasks. Additionally, we included parameterization to simulate different clients and configured JMeter to monitor key performance metrics such as response time, throughput, and error rate.

Strategic testing also includes analysing results to identify performance gaps and ensure the API meets performance objectives. We want to ensure the reliability, capacity, and performance of the API in diverse and complex environments through this comprehensive training using Apache JMeter. Other than that, we have tested the data manually by sending the data through frontend and check it at the database server. We use a very helpful tool named Postman which helped us to see if requests are going on backend properly or not.

4.2 TEST CASES

When testing the performance of our API, we create custom tests using Apache JMeter and adjust the settings to simulate real users. The test plan involves creating a scenario in which 100 virtual users were interact with the API and each user was performing the test twice, 5 seconds apart. This approach is designed to simulate a dynamic user environment by gradually introducing virtual users over a short period of time. The cycle count of 2 ensures that each client repeats the interaction process, providing visibility into API consistency and behaviour across multiple iterations.



Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 100

Ramp-Up Period (in seconds): 5

Loop Count: □ Forever 2

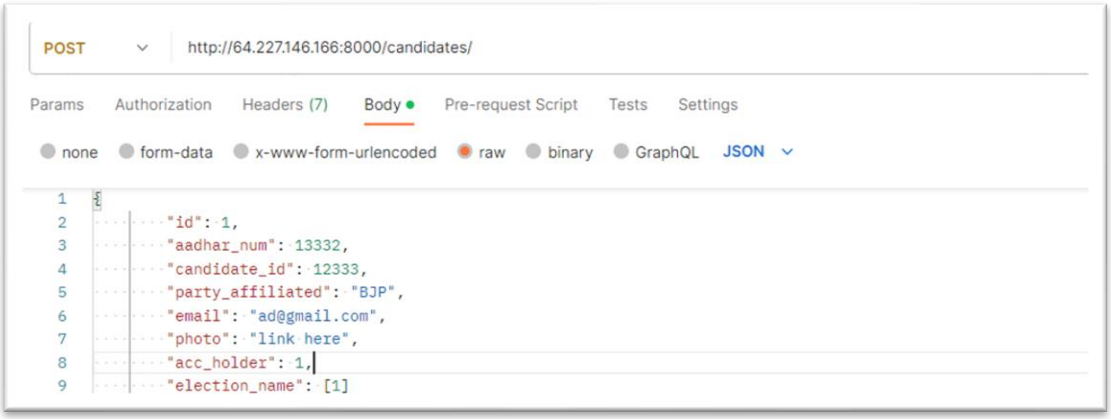
□ Delay Thread creation until needed

□ Scheduler

Figure 4.1: Apache JMeter Testing

This test setup allowed us to test the API's ability to handle concurrent requests, identify potential bottlenecks, and measure performance against average users. Options are balanced between weight and time, providing a comprehensive view of API performance. Analysing the results of these test cases allows us to make informed decisions on how to optimize and improve the reliability and performance of the API in real-world use. Postman helped us to see if requests are going on backend properly or not.

4.3 MANUAL TESTING



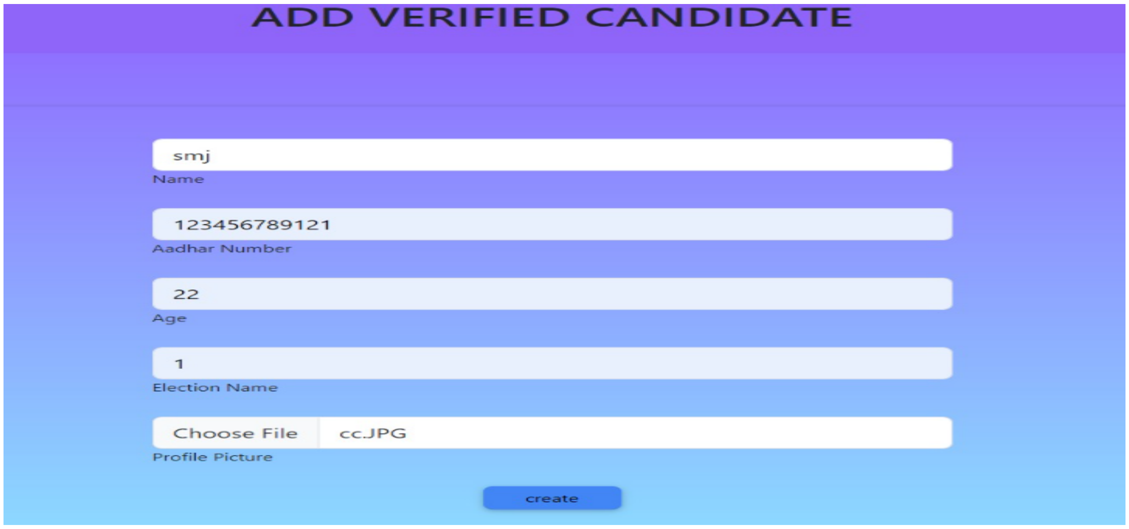
POST http://64.227.146.166:8000/candidates/

● raw JSON

```
{
    "id": 1,
    "aadhar_num": 13332,
    "candidate_id": 12333,
    "party_affiliated": "BJP",
    "email": "ad@gmail.com",
    "photo": "link here",
    "acc_holder": 1,
    "election_name": [1]
```

Figure 4.2: Postman Server Testing

Other than the Apache JMeter and postman, the manual testing for each of the data is also taken place for different validations and speed of the project. Analysing the results of these test cases allows us to make informed decisions on how to optimize and improve the reliability and performance of the API in real-world use.



ADD VERIFIED CANDIDATE

Name: smj

Aadhar Number: 123456789121

Age: 22

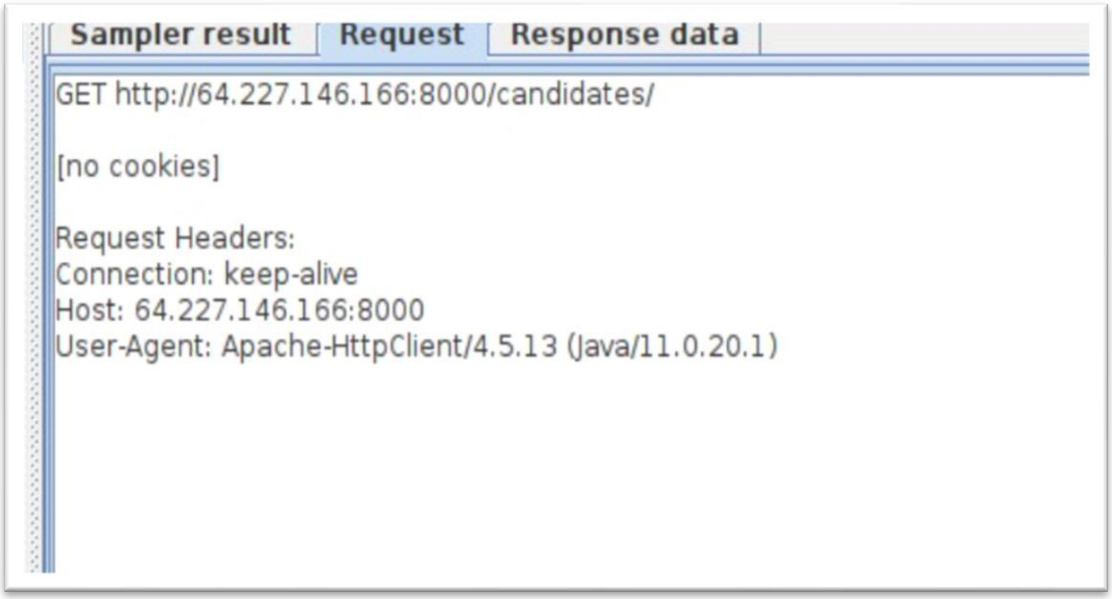
Election Name: 1

Profile Picture: cc.JPG

Figure 4.3: Manual Site Testing

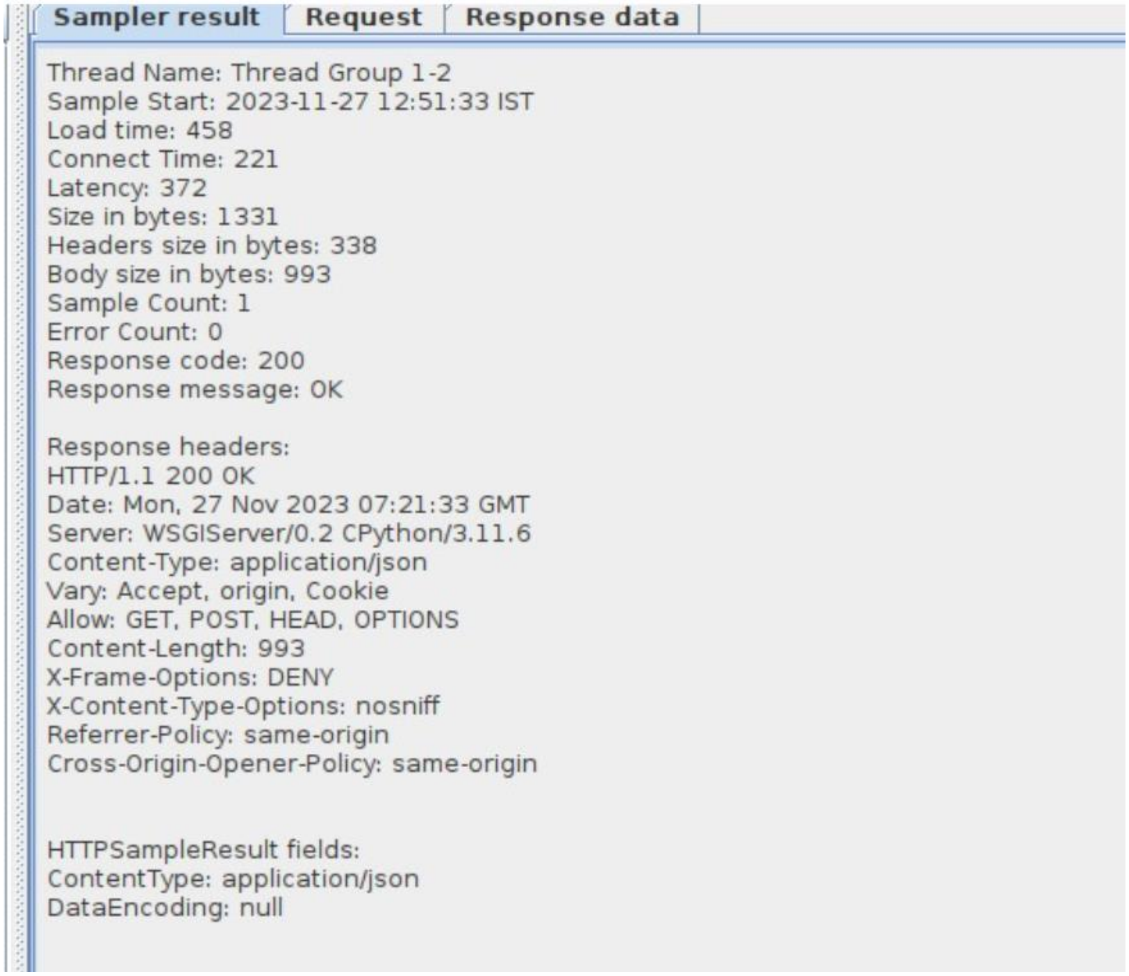
4.4 OUTCOMES

The performance test conducted on the API endpoint "http://64.227.146.166:8000/candidates/" using Apache JMeter produced a successful result with a response code of 200, indicating that the API processed the request without errors. The thread named "Thread Group 1-2" initiated the request, and the sample started on November 27, 2023, at 12:51:33 IST. The overall load time was measured at 458 milliseconds, comprising a connection time of 221 milliseconds and a latency of 372 milliseconds. The response body, encoded as JSON with a content type of "application/json," contained information about candidates. The response headers included details such as the server type (WSGIServer/0.2 CPython/3.11.6), content length (993 bytes), and security-related policies (X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Cross-Origin-Opener-Policy). The test successfully retrieved a list of candidates affiliated with different parties, each represented by a JSON object containing details like ID, Aadhar number, candidate ID, party affiliation, email, photo link, account holder status, and election name association. The absence of errors, coupled with the prompt and accurate response, indicates the API's robustness and functionality under the specified testing conditions.



```
GET http://64.227.146.166:8000/candidates/

[no cookies]

Request Headers:
Connection: keep-alive
Host: 64.227.146.166:8000
User-Agent: Apache-HttpClient/4.5.13 (Java/11.0.20.1)
```

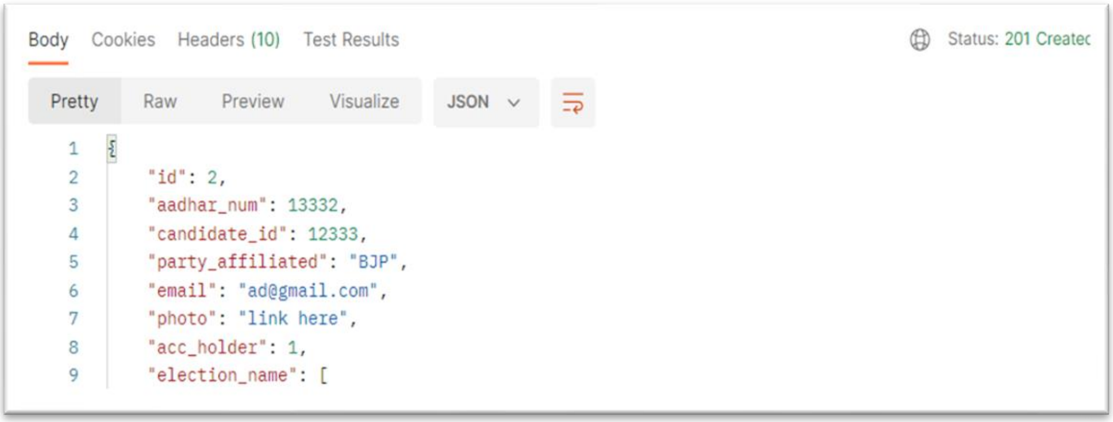
Figure 4.4: Request outcome of Apache JMeter



```
Thread Name: Thread Group 1-2
Sample Start: 2023-11-27 12:51:33 IST
Load time: 458
Connect Time: 221
Latency: 372
Size in bytes: 1331
Headers size in bytes: 338
Body size in bytes: 993
Sample Count: 1
Error Count: 0
Response code: 200
Response message: OK

Response headers:
HTTP/1.1 200 OK
Date: Mon, 27 Nov 2023 07:21:33 GMT
Server: WSGIServer/0.2 CPython/3.11.6
Content-Type: application/json
Vary: Accept, origin, Cookie
Allow: GET, POST, HEAD, OPTIONS
Content-Length: 993
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin


HTTPSampleResult fields:
ContentType: application/json
DataEncoding: null
```

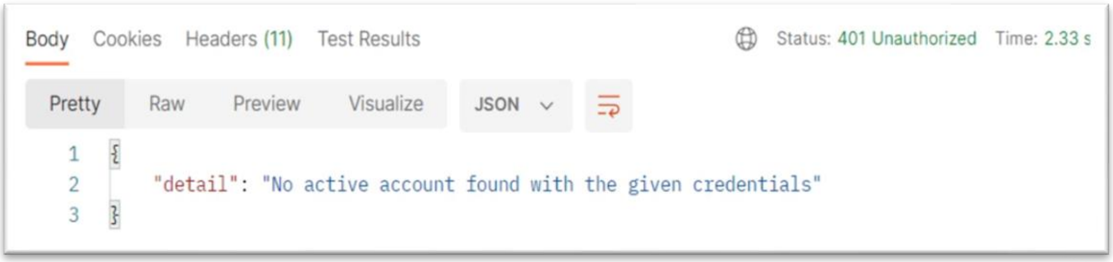
Figure 4.5: Sample Result outcome of Apache JMeter

Also, the performance test conducted on the API endpoint "http://64.227.146.166:8000/candidates/" using postman on producing a successful result given response code of 201, indicating that the API processed the request without errors. While when errors were there, result have given different response code like 400(bad request).



```
{
    "id": 2,
    "aadhar_num": 13332,
    "candidate_id": 12333,
    "party_affiliated": "BJP",
    "email": "ad@gmail.com",
    "photo": "link here",
    "acc_holder": 1,
    "election_name": [
```

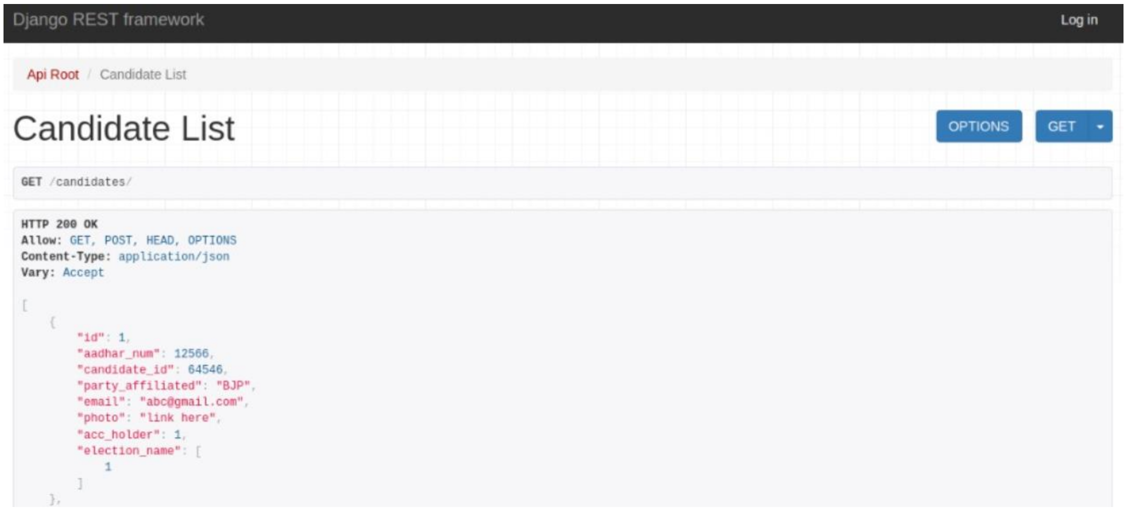
Figure 4.6: Positive response from Postman



```
{
    "detail": "No active account found with the given credentials"
}
```

Figure 4.7: Negative response from Postman

Also, the performance test conducted manually on the endpoint “http://64.227.146.166:8000/candidates/” have given the result and outcomes like this -



```
GET /candidates/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "aadhar_num": 12566,
        "candidate_id": 64546,
        "party_affiliated": "BJP",
        "email": "abc@gmail.com",
        "photo": "link here",
        "acc_holder": 1,
        "election_name": [
            1
        ]
    },
```

Figure 4.8: Result of entered data manually onsite

Chapter 5: RESULTS AND EVALUATION

5.1 PRESENTATION OF FINDINGS

Presenting the findings of a challenge regarding the implementation of visual cryptography for securing online voting calls for a clean and prepared approach. Below is a cautioned structure for offering your findings –

1. **INTRODUCTION TO FINDINGS –** The securing of online voting system using visual cryptography is for securing the voter's confidentiality, integrity and maintain their trust. The object was to find an online voting system that is secured, trustworthy and easy to use.

2. **SECURITY EVALUATION –** The core concept of Visual cryptography has been used in the project to maintain the security. But as visual cryptography is prone to attacks that's why other security features like OTP verification, Aadhar authentication will also be adding.

3. **PRIVACY AND VOTER'S FEEDBACK –** The most important thing in a system like this is trust of people using it, so that's why we will be sending a mail after everything is done in order to take the feedbacks of the voters or we can say users of the system.

4. **USABILITY ANALYSIS –** It is important to make the system that is scalable and won't get delayed or lack if the crown on the site become too high, so for that we have test the system through Apache JMeter.

5. **ACCESSIBILTY EVALUATION –** It is very important that the presented system or solution is accessible to everyone, but as the site doesn't support to those who are visually disable, we will be seeing into to issue in future scope.

5.2 INTERPRETATION OF THE RESULT

Interpretation of the result means the understanding of the findings so that the securing of voting system using visual cryptography can be conducted smoothly without any issue of security, lost of trust or confidentiality of the user. So, some of the interpretation of the result across various aspects of the project are –

1. **SECURITY EVALUATION** – A successful security evaluation tells how well the person is able to secure and safeguard the voting system. A system with least vulnerabilities and effective strategies towards the solution have the best security evaluation.

2. **PRIVACY AND VOTER'S FEEDBACK –** Positive feedbacks from the users say that they are happy with the process and it is effective in terms of saving their private information and maintaining their confidentiality and integrity and most importantly, their trust.

3. **USABILITY ANALYSIS** – Improved usability analysis of the online voting system tells that the system has not compromised the user's experience. Positive users number means that system is working properly and nicely with full of their trust.

4. **ACCESSIBILTY EVALUATION –** The more the system is accessible, the better the chances are for the growth of the system. It must be accessible to each location, to each kind of people without any discrimination based on their caste, color, creed or physical condition.

5. **SCALABILITY ASSISSMENT –** Positive scalability results indicate that the online voting system, with visual cryptography, can handle an increased volume of votes efficiently. This is critical for ensuring the system's reliability during periods of high demand.

5.3 WHAT WE HAVE ACHIEVED

Visual cryptography can offer several potential results when used in voting systems. Some important results that it offers and we have gained are discussed below:

1. **ENHANCED SECURITY** – By making it more difficult for attackers to manipulate or steal sensitive information, such as voting preferences or voter registration information, visual cryptography helped us to improve the security of voting systems. Voting options can be represented visually in distinctive ways that are challenging for attackers to duplicate or manipulate.

2. **INCREASED TRANSPARENCY** – Voters can view a visual depiction of their selections and confirm that their vote has been accurately recorded by employing visual cryptography, which increases voting transparency. This may contribute to a rise in public confidence in the electoral process.

3. **IMPROVED EFFICIENCY** – The use of visual cryptography can increase voting system effectiveness by obviating the requirement for manual vote counting and verification. Votes may be simply and rapidly confirmed with visual cryptography, which can hasten the voting process.

4. **RESISTANCE TO ATTACKS** – By making it more difficult for attackers to intercept or manipulate voting data, visual cryptography can increase the security of voting systems. The use of visual cryptography can aid in preventing phishing attacks, which aim to modify or steal important data and maintain the fairness of the voting process.

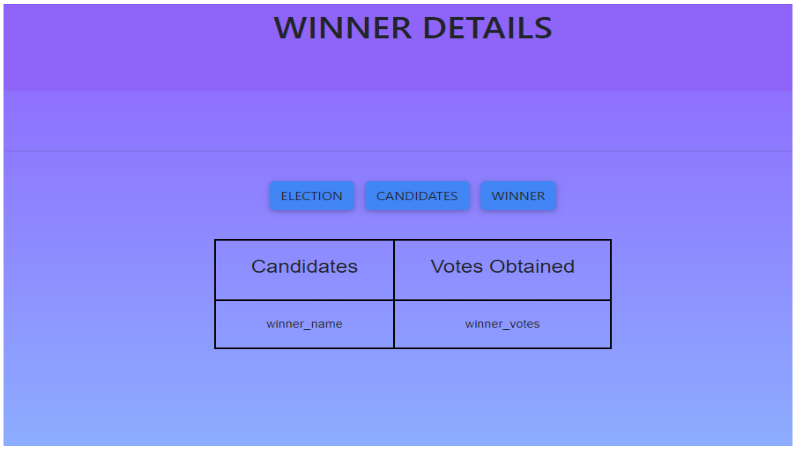
5. **PROTECTION OF VOTER'S PRIVACY** – By making it as a system that is fully secured and in which users can vote without fear of losing their privacy. Only those who have access to the results that is admin can see the result and that is only their name, nothing else.

6. **APPLICABILITY TO DIFFERENT VOTING SYSTEMS** – There are many kinds of voting systems that can use visual cryptography, including electronic voting systems, online voting systems, and paper-based voting systems.

7. **VERIFICATION OF VOTER'S INTEGRITY** – Using visual cryptography, voters can check that the recorded decision corresponds to their intention by viewing a visual depiction of their selections.

8. **ACCESSIBILTY** – It is possible to build visual cryptography so that it is usable by voters with disabilities or vision impairments. Visual cryptography may be developed to meet the various needs of voters by employing the right methods and tools.

The use of visual cryptography in voting systems has substantial potential outcomes overall. Visual cryptography can help to ensure that the election process is fair, accurate, and trustworthy by enhancing security, increasing transparency, improving efficiency, and making voting systems more resistant to attacks.



Candidates	Votes Obtained
winner_name	winner_votes

Figure 5.1: Result of winner Page

In the result section, apart from the table that shows about the votes each candidate get, we also have a pie chart and bar graph to show the result of the voting. Pie charts and bar graphs give data a visual representation that helps viewers quickly understand how votes were distributed among the candidates. Using visual aids improves the presentation as a whole and increases the readability and engagement of the results.

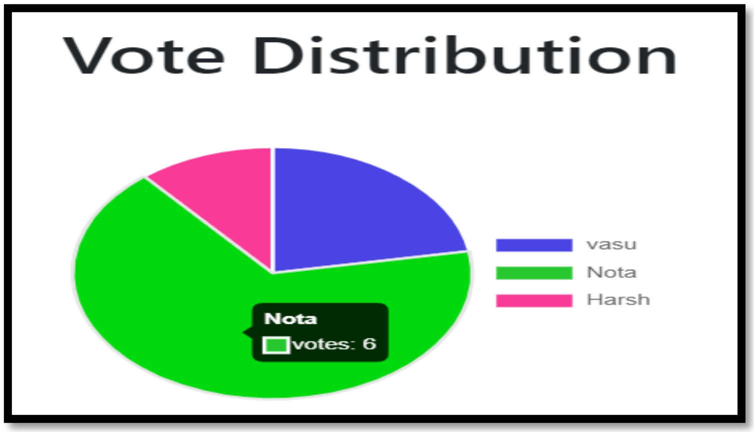


Figure 5.2: Result pie chart

Bar graphs make it simple to compare how many votes each candidate received. Viewers can compare the vote totals for various candidates and quickly determine which one received the most votes. This comparison facilitates the analysis of each candidate's support and popularity.

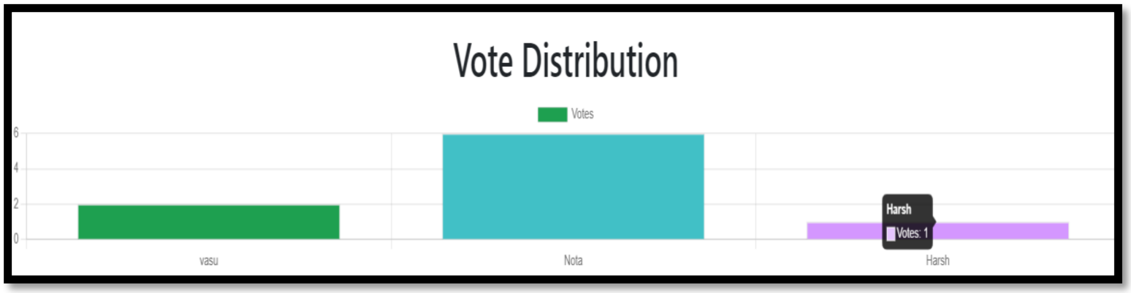


Figure 5.3: Result Bar Graph

Pie charts show the percentage distribution of votes among the contenders, indicating the share of votes that each candidate received in relation to the total number of votes. The percentage of the vote that each candidate received and their individual contributions to the final outcome are effectively displayed by this graphic.

The presentation gains depth and variety when it employs various presentation formats for the results. Pie charts show proportions, whereas bar graphs show absolute values. This diversity facilitates the effective dissemination of important discoveries and insights to audiences and stakeholders. A broad range of audiences, including those with little experience with data analysis, can easily understand visualizations like pie charts and bar graphs. To reach a wide audience and clearly communicate the voting results, they can be included in reports, presentations, or online platforms.

All things considered, using various forms, like pie charts and bar graphs, improves the data analysis and comprehension by improving the visual impact, communication, and clarity of the voting results.

Chapter 6: CONCLUSIONS AND FUTURE SCOPE

6.1 APPLICATIONS

We have already seen that how using visual cryptography is beneficial for securing the online voting system but now we can see other applications of using visual cryptography as our security model –

1. **Secure Communication** – Visual cryptography can be used where some sensitive visual data is getting transfer from one end to another. The data can be shared in two or more parts and get superimposed at receiver end so that it can't be stole in the middle.

2. **Digital Watermarking** – Visual cryptography can be used for digital watermarking where text will get faded under black and white shattered images and would be shown only when the images are superimposed.

3. **Secure Cloud Storage** – Visual cryptography can be very useful to secure cloud as the sensitive data can be transferred very securely in different shares where the original sequence is known by the authorized users only.

4. **Biometric template protection** – Visual cryptography can be used to protect the template of biometric so that the data will not be shown only in a single location and can secure the confidentiality of users.

These applications show how versatile this visual cryptography algorithm is and how better it can work to maintain security, integrity and confidentiality of the users.

6.2 LIMITATIONS

Although we have given our best to the project but still there are some limitations and issues in the project which we will be trying to solve in the future with our full mind and soul to the project –

1. **Accessibility** – This model is not accessible to all kind of users, specially those who are visually blind or can’t see as we have not added any voice assistant in the model, which can be our future scope for the project.

2. **Voter Education** – As the model is fully English oriented and as we know if the people from different places and ground want to vote, the model must be present in different languages so that people can vote easily without any problem.

3. **Scalability** – It is a testing model that’s why when the number of voters become too high, it starts becoming difficult to maintain the process smoothly without any delays or lacks.

4. **Bounded** – This is a testing model and will run only for one admin at one time, so if more than one admin come into the picture, it will get stuck. This is the focus for us in the future scope.

5. **Trust in technology** – Voters need to trust the visual cryptography and believe that their votes are secured. They must believe that the concept is made just to maintain their votes security, integrity and confidentiality.

6.3 CONCLUSION

In conclusion, securing online voting system using visual cryptography is a very potential and promising approach in terms of securing the voter's privacy, integrity and confidentiality, which includes –

- Securing online Voting system shows that it is possible to provide a secure voting service using Internet with full confidentiality, integrity, security and privacy of the user.

- Secure online Voting system has potential to reduce expenses for elections. This include both money and human resource. This includes all the cost of election commission, workers, labors and time of all the officers and officials.

- Fair and limited use of resources by this system proves it's efficiency. Secure online Voting system has potential to conduct fair elections where voters can vote without any fear of loss of their identity and personal details.

- Secure online Voting system provided elegant yet easy to use voting service. Tutorials and FAQs provided with the system tries to answers most queries asked by users. Result of "User acceptance" testing support these conclusions.

- The scope of this system is limited to elections within an organization. Using this system in large elections like elections in state/country may cause scalability issues but this can be used in that area as well with working on the scalability issue for the future scope.

6.4 FUTURE WORK

Although the basic model of the project is ready but there are many scopes of improvement for future and here are some potential future works for visual cryptography in online voting systems –

1. **Blockchain Technology –** Blockchain technology can be used for the security of database. Blockchain technology provides a secure and transparent way to record and transfer data. Its decentralized nature, encryption, transparency, immutability, and smart contracts all contribute to resolving data privacy and security issues for businesses

2. **Scalability –** The scalability of secure E-Vote is also a major concern. We would like to improve scalability of this system so that it can be used in large scale elections without any kind of delay or lacks. To improve scalability, we are thinking of using client-side computation.

3. **Strengthening security -** As was already established, visual cryptography is prone to attacks, so the system's security needs to be increased. So, there are many other security techniques are also present in the market which help us to secure the system like AES, DES, Biometric censors, Homomorphic algorithm etc

4. **Add Payment System –** For the future, adding payments is also one of the plans that we are thinking of, as payment also need to be secured and if this model would be used by organizations that to make the project or model as earning model, it must be reasonable so that it would be the first priority for the organizations.

REFERENCES

1. M. Ardakan, R. Ramezani, A. Latif, “Visual Secret Sharing of Gray and Color Images using Fuzzy Random Grids”, Applied Soft Computing, Vol. 146, Oct 2023.

2. V. Anitha, O. Caro, R. Sudharsan, S. Yoganandan, M. Vimal, “Transparent voting system using blockchain”, Journal of the International Measurement Confederation, Vol. 25, Feb 2023.

3. W. Xiaotian, C. Yang, H. Cai, Y. Liu, “A hybrid approach combining data hiding with visual cryptography for secure extraction of data hiding”, Journal of Information Security and Applications, Vol. 75, Jun 2023.

4. A. Abapour, N. Navid, M. Ebadpour. “Decentralized lattice-based method for visual symmetric cryptography”, Franklin Open, Vol. 3, Jun 2023.

5. G. Tiwari and K. Kumar, “Secure online voting system using visual cryptography: a comparative analysis”, Walailak Journal of Science and Technology, Vol. 18, Jul 2021.

6. S. Sridhar, G. Sudha, “progressive visual cryptography using simple modular arithmetic operations”, Journal of visual information and image communication, Vol. 74, Apr 2021.

7. J. GwangSu, U. Ko, “Research on a novel construction of probabilistic visual cryptography scheme (k, n, 0, 1, 1) − PVCS for threshold access structures”, Theoretical Computer Science, Vol. 863, Apr 2021.

8. S. Saproo, V. Warke, S. Pote, R. Dhumal, “Online voting system using Homomorphic encryption”, ITM web of conferences, Vol. 32, Dec 2020.

9. A. Dwivedi, G. Khandare, M. Shaikh, A. Morey, “Online Voting system based on visual cryptography and biometric”, International Research Journal of Engineering and Technology, Vol. 08, June 2021.

10. W. Patel, M. Patel, B. Ramani, “A review of online voting system security based on cryptography”, International Conference on Smart Data Intelligence Vol. 09, May 2021.

11. Y. Shah, R. Rane, S. Kharade, R. Patil, “Analysis of AES and DES algorithm”, International Journal of Trend in Research and Development, Vol. 07, April 2020.

12. G. Teng, F. Liu, C. Wu, “K out of k extended visual cryptography scheme by random grids”, Signal Processing, Vol. 94, Jan 2016.

13. M. Naor, M. Moni, and A. Shamir, "Visual cryptography", Workshop on the Theory, Vol. 12, May 1995.

14. S. Aggarwal, "Modern web-development using reactjs.", International Journal of Recent Research, Vol. 1, May 2018.

15. B. Burch, J. Carl. "Django, a web framework using python: Tutorial presentation.", Journal of Computing Sciences, Vol. 7, Jun 2010.