Design and Implementation of Online Voting System

by

Md Eleus
ID: CSE1901016161

Md. Rakibul Hasan
ID: CSE2001019040

Nasir Uddin
ID: CSE1901016167

Supervised by
Mohammad Naderuzzaman

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
March 2023

APPROVAL

The project titled "**Design and Implementation of Online Voting System**" submitted by Md Eleus (CSE1901016161), Md. Rakibul Hasan (CSE2001019040) and Nasir Uddin (CSE1901016167) to the Department of Computer Science and Engineering, Sonargaon University (SU), have been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Mohammad Naderuzzaman — **Supervisor**
Associate Professor,
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 1**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 2**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 3**
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Mohammad Naderuzzaman, Associate Professor,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned	Signature
----------	----------
(Mohammad Naderuzzaman)	Md Eleus
Supervisor	ID: CSE1901016161

	Md. Rakibul Hasan
	ID: CSE2001019040

	Nasir Uddin
	ID: CSE1901016161

ABSTRACT

An online voting system is a web-based application that enables eligible voters to cast their votes via the internet. The system typically involves a website that presents the ballot to the voter, verifies the voter's identity, and records the vote. Online voting systems have become increasingly popular in recent years due to their convenience, accessibility, and potential to increase voter turnout. However, there are concerns about the security and integrity of online voting systems, including the risk of hacking, vote manipulation, and voter impersonation. To address these concerns, online voting systems typically incorporate a range of security measures, such as encryption, authentication, and audit trails. The development and implementation of online voting systems require careful planning, testing, and evaluation to ensure their effectiveness and security.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to Almighty Allah for giving us the ability and strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Mohammad Naderuzzaman**, Associate Professor, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are thankful to **Prof. Bulbul Ahamed**, Professor & Head, Department of Computer Science and Engineering, for approving our project.

We would like to convey our special gratitude to **Brig. Gen. (Retd) Prof. Habibur Rahman Kamal, NDC, PSC** Dean, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
CDN	Content Delivery Network
Docker	Containerization Platform
DOM	Document Object Model
Git	Distributed Version Control System
HTML	HyperText Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
JWT	JSON Web Token
JSON	JavaScript Object Notation
JS	JavaScript
MERN	MongoDB, Express.js, React, Node.js
NoSQL	Not Only SQL
Npm	Node Package Manager
OOP	Object-Oriented Programming
OAuth	Open Authorization
ORM	Object-Relational Mapping
PaaS	Platform as a Service
REST	Representational State Transfer
SASS	Syntactically Awesome Stylesheets
SEO	Search Engine Optimization
SSO	Single Sign-On
SQL	Structured Query Language
SPA	Single Page Application
SSL	Secure Sockets Layer
TDD	Test-Driven Development
TLS	Transport Layer Security
UI	User Interface
UX	User Experience
VM	Virtual Machine
HTTPS	Hypertext Transfer Protocol Secure
JWT	JSON Web Token
SASS	Syntactically Awesome Stylesheets
SEO	Search Engine Optimization
SSL	Secure Sockets Layer

TLS	Transport Layer Security
URL	Uniform Resource Locator
VM	Virtual Machine

TABLE OF CONTENTS

Title		Page No.
DECLARATION ……..……..		iii
ABSTRACT ……...…….		iv
ACKNOWLEDGEMENT …….……		v
LIST OF ABBREVIATION …….…….		vi
CHAPTER 1		1 – 6
INTRODUCTION OF ONLINE VOTING SYSTEM		
1.1	Introduction ……..…….....	1
1.2	Objectives ……...……...	1-2
1.3	Proposed ……..…….	3-4
1.4	Benefit …….…….....	4-5
1.5	Overview …….…….	5-6
CHAPTER 2		7-9
BACKGROUND STUDY		
2.1	Discussion With Client…….…….…….	7
2.2	Discussion With Supervisor……...	7-8
2.3	Web Search ……...	8
2.4	Using MERN Stack……	8-9
CHAPTER 3		10-14
METHODOLOGY		
3.1	Objective……..….....	10
3.2	System Analysis……...…….…….	10
3.3	Values, Principles……...	10-11
3.4	Reason For Using Agile Methodology ……...	11-12
3.5	Advantages Of Agile Method ……	12-13
3.6	Disadvantages Of Agile …….	13-14
CHAPTER 4		15-20
SYSTEM DESIGN		
4.1	Requirement Specification……..…….....	15-16
4.2	System Flowchart……..…….	17-20

CHAPTER 5 21-42

IMPLEMENTATION AND OUTCOME

5.1 Tools and Technology Used for Development …….. 21-30

5.2 Software ……. 31

5.3 Hardware 31

5.4 Database Service …… 32

5.5 Implementation…… 32-33

5.6 Public View …… 34-37

5.7 Candidate View …….. 38

5.8 Admin View ……... 38-42

CHAPTER 6 43-44

CONCLUSION AND FUTURE WORKS

6.1 Discussion……..……. 43

6.2 Scope for Further Developments ……..……. 43-44

6.3 Conclusion ……..…… 44

REFERENCES …….……. 45

LIST OF TABLES

Table No.	Title	Page No.
Table 5.1	Software Used For Development	31
Table 5.2	Hardware Used For Development	31

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.2.1	Group Discussion	7
Fig.2.2	Web Search	8
Fig.2.3	MERN Stack	9
Fig.3.1	Agile Model	11
Fig.4.1	User Flow	17
Fig.4.2	Candidate Flow	18
Fig.4.3	Admin Flow	19
Fig.4.4	ER Diagram	20
Fig.5.1	MongoDB	21
Fig.5.2	NodeJs+Express js	22
Fig.5.3	ReactJs	24
Fig.5.4	NodeJs	26
Fig.5.5	Testing Framework	28
Fig.5.6	Home Page	34
Fig.5.7	About Page	35
Fig.5.8	Result Page	35
Fig.5.9	Voter Login	36
Fig.5.10	Candidate Login	36
Fig.5.11	Voter Register	37
Fig.5.12	Candidate Register	37
Fig.5.13	Give Vote	38
Fig.5.14	Admin Login	38
Fig.5.15	Admin Dashboard	39
Fig.5.16	Appellation List	39
Fig.5.17	Pending Candidate	40
Fig.5.18	Confirm Candidate	40
Fig.5.19	Confirm Voter	41
Fig.5.20	Pending Voter	41
Fig.5.21	Election Result	42

CHAPTER 1

INTRODUCTION OF ONLINE VOTING SYSTEM

1.1 Introduction

An online voting system revolutionizes the traditional methods of casting votes by leveraging digital technology to streamline and enhance the electoral process. This system allows voters to securely and conveniently participate in elections from any location with internet access. Its introduction marks a significant advancement in democratic practices, offering numerous benefits such as increased accessibility, efficiency, and accuracy in electoral procedures.

At its core, an online voting system typically comprises a user-friendly interface accessible via web browsers or dedicated applications. Through this interface, eligible voters can authenticate their identity securely, ensuring the integrity of the electoral process. Once authenticated, voters can browse candidate profiles, review ballot measures, and cast their votes electronically.

The implementation of robust security measures is paramount in safeguarding the integrity and confidentiality of online voting. Encryption protocols, multi-factor authentication, and tamper-evident audit trails are commonly employed to mitigate potential risks such as hacking, tampering, or identity fraud. Additionally, stringent verification procedures and regular security audits are conducted to uphold the credibility and trustworthiness of the system.

Online voting systems offer unparalleled convenience by eliminating the constraints of physical polling locations and extended voting hours. Voters can participate in elections at their convenience, reducing barriers such as long queues, transportation issues, or scheduling conflicts. This accessibility promotes greater voter turnout and engagement, fostering a more inclusive and representative democratic process.

Furthermore, online voting enhances the efficiency and accuracy of election administration. Automated tabulation and instantaneous result reporting expedite the vote-counting process, minimizing delays and uncertainties associated with manual tallying. Real-time monitoring capabilities enable electoral authorities to promptly address any irregularities or discrepancies, ensuring the fairness and transparency of the electoral outcome.

Despite its potential advantages, the introduction of an online voting system requires careful consideration of legal, logistical, and technological factors. Regulatory frameworks must be established to address issues such as voter eligibility, privacy protection, and dispute resolution. Robust infrastructure and contingency plans are essential to mitigate the risk of technical failures or cyber-attacks. Additionally, comprehensive public education campaigns are necessary to familiarize voters with the new system and build trust in its reliability and security.

1.2 Objectives

The objectives of an online voting system typically revolve around improving various aspects of the electoral process, enhancing democracy, and addressing existing challenges associated with traditional voting methods.

Ensure that all eligible voters, regardless of their location or physical abilities, have easy and convenient access to the voting process. Online voting eliminates barriers such as transportation issues, long queues, and geographical constraints, thereby promoting greater voter participation and inclusivity.

Offer voters a user-friendly and efficient voting experience that accommodates their busy schedules and personal commitments. Online voting enables voters to cast their ballots from any internet-enabled device at their convenience, reducing the need for physical presence at polling stations and long waiting times.

Implement robust security measures to safeguard the integrity, confidentiality, and authenticity of the voting process. Utilize encryption techniques, multi-factor authentication, and tamper-evident audit trails to protect against cyber threats, hacking attempts, and fraudulent activities, thereby ensuring the credibility and trustworthiness of election results.

Enhance the accuracy and reliability of the electoral process by minimizing errors, discrepancies, and delays associated with manual vote counting and tabulation. Automated tabulation systems and real-time monitoring capabilities enable swift and accurate reporting of election results, reducing the likelihood of human error and ensuring the transparency of the electoral outcome.

Foster transparency and accountability in the electoral process by providing stakeholders, including voters, candidates, and electoral authorities, with visibility into the voting process and its outcomes. Implement open-source voting software, public audits, and verifiable voting protocols to promote trust and confidence in the integrity of the electoral system.

Streamline election administration and reduce operational costs associated with traditional voting methods, such as printing paper ballots, staffing polling stations, and transporting voting equipment. Online voting systems offer a cost-effective alternative by automating various aspects of the electoral process and optimizing resource allocation.

Ensure that the online voting system is adaptable to evolving technological advancements, regulatory requirements, and user preferences. Regular updates, system enhancements, and stakeholder feedback mechanisms enable continuous improvement and innovation, thereby maintaining the relevance and effectiveness of the voting system over time.

Encourage civic engagement and democratic participation by providing opportunities for voter education, outreach, and empowerment. Public awareness campaigns, voter registration drives, and civic education initiatives aim to inform and mobilize citizens to exercise their right to vote and actively participate in the democratic process.

By aligning with these objectives, an online voting system can contribute to the advancement of democratic principles, promote electoral integrity, and enhance the overall efficiency and accessibility of the electoral process.

1.3 Purpose

The purpose of an online voting system is multifaceted, encompassing several overarching objectives aimed at modernizing and improving the electoral process.

One of the primary purposes of an online voting system is to increase accessibility to the voting process. By allowing voters to cast their ballots from any location with internet access, online voting removes barriers such as physical disabilities, distance from polling stations, and time constraints. This increased accessibility promotes greater voter turnout and ensures that all eligible citizens have the opportunity to participate in elections.

Online voting systems aim to make the voting process more convenient for voters. With the ability to vote from the comfort of their own homes or on the go using internet-enabled devices, voters no longer need to take time off work or travel to polling stations. This convenience encourages higher voter participation rates and makes it easier for busy individuals to engage in the democratic process.

Another purpose of online voting systems is to streamline the electoral process and make it more efficient. Automated tabulation of votes, instant result reporting, and reduced reliance on manual processes help expedite the counting and reporting of election results. This efficiency minimizes delays and uncertainties associated with traditional voting methods, ensuring a smoother and more timely electoral process.

Online voting systems prioritize security measures to safeguard the integrity and confidentiality of the voting process. Utilizing encryption, authentication protocols, and tamper-evident audit trails, these systems aim to protect against cyber threats, hacking attempts, and fraud. By implementing robust security measures, online voting systems ensure that election results are accurate, trustworthy, and free from manipulation.

Implementing online voting systems can lead to cost savings for electoral authorities. By reducing the need for printed ballots, physical polling stations, and manual labor associated with traditional voting methods, online voting systems help lower the operational costs of conducting elections. This cost efficiency allows electoral authorities to allocate resources more effectively and invest in other critical areas of electoral administration.

Online voting systems promote transparency in the electoral process by providing stakeholders with visibility into the voting process and its outcomes. Features such as open-source voting software, public audits, and verifiable voting protocols help build trust and confidence in the integrity of the electoral system. By enhancing transparency, online voting systems ensure that election results are credible and accountable to the public.

As technology continues to evolve, online voting systems aim to remain adaptable and responsive to emerging trends and innovations. Regular updates, system enhancements, and stakeholder feedback mechanisms ensure that online voting systems stay current with evolving technological standards and regulatory requirements. This adaptability helps maintain the relevance and effectiveness of online voting systems in an ever-changing digital landscape.

The purpose of an online voting system is to modernize the electoral process, increase voter participation, improve efficiency and security, and uphold the principles of democracy in the digital age. By achieving these objectives, online voting systems contribute to a more accessible, convenient, and trustworthy electoral experience for citizens around the world.

1.4 Benefit

The benefits of an online voting system are numerous and encompass various aspects of the electoral process, democratic governance, and societal engagement.

Online voting systems enable voters to participate in elections from any location with internet access, eliminating barriers such as distance from polling stations, mobility issues, and time constraints. This increased accessibility promotes greater voter turnout and ensures that all eligible citizens have the opportunity to exercise their democratic rights.

Online voting offers voters the convenience of casting their ballots remotely, without the need to travel to physical polling stations or wait in long queues. Voters can participate in elections at their convenience, whether from home, work, or while traveling, leading to higher levels of voter engagement and participation.

Online voting systems can lead to significant cost savings for electoral authorities by reducing the need for printed ballots, physical polling stations, and manual labor associated with traditional voting methods. This cost efficiency allows electoral authorities to allocate resources more effectively and invest in other critical areas of electoral administration.

Automated tabulation of votes, instant result reporting, and reduced reliance on manual processes streamline the electoral process, making it more efficient and timely. Online voting systems expedite the counting and reporting of election results, minimizing delays and uncertainties associated with traditional voting methods.

While ensuring the security of online voting systems is paramount, when properly implemented, they can offer robust security features such as encryption, authentication protocols, and tamper-evident audit trails. These measures help safeguard the integrity and confidentiality of the voting process, protecting against cyber threats, hacking attempts, and fraud.

Online voting systems promote transparency in the electoral process by providing stakeholders with visibility into the voting process and its outcomes. Features such as open-source voting software, public audits, and verifiable voting protocols help build trust and confidence in the integrity of the electoral system.

By reducing the need for paper-based ballots and physical infrastructure associated with traditional voting methods, online voting systems contribute to environmental sustainability. They help minimize the environmental impact of elections by conserving resources and reducing carbon emissions associated with printing and transportation.

Online voting systems can adapt to evolving technological trends and innovations, ensuring that they remain relevant and effective in the face of changing digital landscapes. Regular

updates, system enhancements, and stakeholder feedback mechanisms help maintain the security, accessibility, and usability of online voting systems over time.

The benefits of online voting systems include increased accessibility, convenience, cost savings, efficiency, security, transparency, environmental sustainability, and adaptability to technological advancements. By leveraging digital technology to modernize the electoral process, online voting systems contribute to a more accessible, inclusive, and trustworthy democratic society.

1.5 Overview

An online voting system is a digital platform that enables eligible voters to cast their ballots remotely via the Internet. It represents a modernized approach to traditional voting methods, leveraging digital technology to streamline and enhance the electoral process. Here's an overview of how online voting systems typically function:

- User Authentication: The online voting system begins with user authentication to verify the identity of voters. Various authentication methods may be employed, including passwords, biometrics, or multi-factor authentication, to ensure that only eligible voters are granted access to the voting platform.
- Voter Registration: Before participating in an online election, voters typically need to register on the voting platform. This registration process may involve verifying voter eligibility, providing personal information, and creating a secure account.
- Ballot Presentation: Once authenticated, voters are presented with the ballot electronically. The online ballot mirrors the traditional paper ballot, listing candidates for various offices, ballot measures, and any other relevant voting options.
- Voting Process: Voters make their selections by electronically marking their choices on the ballot. The interface of the online voting system is designed to be user-friendly, allowing voters to navigate the ballot easily and review their selections before finalizing their votes.
- Security Measures: Online voting systems employ various security measures to safeguard the integrity and confidentiality of the voting process. These measures may include encryption techniques, secure connections, audit trails, and robust authentication protocols to prevent unauthorized access, tampering, or manipulation of votes.
- Vote Submission: After completing the ballot, voters submit their votes electronically through the online voting platform. Once submitted, the votes are securely stored and encrypted to maintain the anonymity of the voters and the integrity of the electoral process.
- Vote Tabulation and Result Reporting: The online voting system automatically tabulates the votes once the voting period ends. Results are generated and reported in real-time, providing immediate visibility into the outcome of the election. Electoral authorities can access comprehensive reports and analytics to analyze voter turnout, demographics, and other relevant data.

- Post-Election Audits and Verification: Following the election, post-election audits may be conducted to verify the accuracy and integrity of the results. These audits may involve comparing electronic records with paper records, conducting security reviews, and investigating any discrepancies or irregularities.
- Accessibility and Inclusivity: Online voting systems enhance accessibility and inclusivity by allowing voters to participate in elections from any location with Internet access. This accessibility reduces barriers such as physical disabilities, distance from polling stations, and time constraints, promoting greater voter turnout and engagement.
- Continuous Improvement and Adaptability: Online voting systems are continually evolving to adapt to technological advancements, regulatory requirements, and user feedback. Regular updates, system enhancements, and stakeholder engagement ensure that online voting systems remain secure, reliable, and user-friendly over time.

CHAPTER 2

Background Study

2.1 Discussion With Client



Fig: 2.1 Discuss with client

Before making this project we had to discuss with some teachers that what is their requirement? Why do they need this site, what are the features they only want? Good client communication often involves actively listening to your customers to interpret their emotions and better cater to their needs Being transparent with a client helps facilitate trust Customers may appreciate learning more about the production process so they can shop consciously We did this thing only because there are so many Website at the market for sell but those are may have more feature that for the school of suburb area and those will cost them much more where they don't want to invest Client communication is important because it establishes and maintains trust between the client and the business When customers trust a brand, they may be more likely to remain loyal Open client communication can also help limit misunderstandings, lead to greater customer satisfaction and make clients more likely to recommend a business to others So this was crucial for us to take this project though this had been created before so many time Having adaptable solutions for communicating can strengthen the relationship between the client and us When a change occurs, flexibility helps both parties continue to communicate The clients are eager to implement this system at their schools for save time management if it costs in a budget range like 20 to 30 thousands BDT Also they want this thing in a very simple way that they can use So This step of work were fruitful for this project

2.2 Discussion With Supervisor

Effective communication with your supervisor is a key element of your eventual success in the workplace Therefore, it is important that you feel comfortable and prepared to approach your supervisor promptly to discuss concerns, request assistance, or report work progress As we have an amazing supervisor to our group we had discussed with him about our simple concept he is good at communication he is clear to her speech, and never did underestimate

us he offered us by brief discussion By that way we were able to take our decision Through some discussion we were able to keep this as our final proposal We didn't choose any thesis like project because

1. We never created a complete project before
2. We had some lack of study before
3. We did not our semester courses during this period

Despite all of the obstacles our supervisor **Mohammad Naderuzzaman**, Associate Professor of Sonargaon University helped us to do the least job that was in front of us now he is an amazing teacher as he never scolded us rather he inspired all of us to do the least basic of the job as this is our first project.

2.3 Web Search



Fig: 2.2 Google Search

Searching through websites is very important nowadays On the web there is almost everything Every project is built with three or more Technologies There needs to be more than one language and a server need to connect there So, as we mentioned earlier that we had the lack of knowledge to build a complete website So we had to see some website and inspected here code from browser After this we had to see some tutorials from youtube how they do the job for a project and read some content from some site.

2.4 Using MERN

The stack of the project is MERN we mainly use React, Express, MongoDB, Nodejs

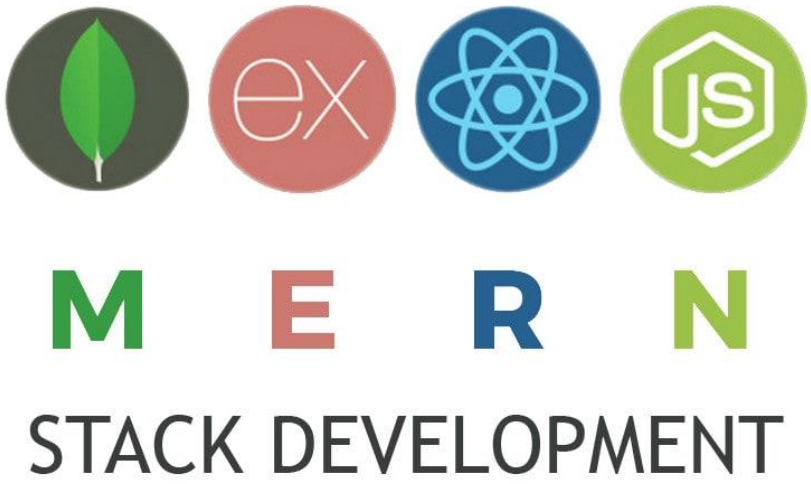


Fig: 2.3 MERN Stack

MEAN is a free and open-source JavaScript software stack for building dynamic websites and web applications. A variation known as MERN replaces Angular with React. is a collection of technologies that enables faster application development. It is used by developers worldwide. The main purpose of using the MERN stack is to develop apps using JavaScript only. This is because the four technologies that make up the technology stack are all JS-based. Thus, if one knows JavaScript (and JSON), the backend, frontend, and database can be operated easily.

CHAPTER 3

METHODOLOGY

3.1 Objective

Methodology refers to the overarching strategy and rationale of your research project It involves studying the methods used in your field and the theories or principles behind them, in order to develop an approach that matches your objectives The methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated Methodology does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective A methodology does not set out to provide solutions it is therefore, not the same as a method Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practices can be applied to a specific case, for example, to calculate a specific result There are various types of methodology Online However, we choose to use AGILE method for our project Online Voting System

3.2 System Analysis

Systems analysis is a process of looking at end-user implementation of a software package or product; looking in-depth at source code to define the methodologies used in building software; or taking feasibility studies and other types of research to support the use and production of a software product, among other things methodologies used in building software; or taking feasibility studies and other types of research to support the use and production of a software product, among other things System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recorded improvements on the system It is a problem solving activity that requires intensive communication between the system user and system developer System analysis or study is an important phase of any system development process A detailed study of the process must be made by various techniques like interviews, questionnaires etc The data collected by these sources must be scrutinized to arrive to conclusion every synthesis is built upon the results of a preceding analysis and every analysis requires a subsequent

3.3 Values, Principles

There are some specific values and principles in Agile Method Those are given Below:

Values:

1. Working software over comprehensive documentation
2. Individuals and interactions over processes and tools
3. Customer collaboration over contract negotiation
4. Responding to change by following a plan

Principles:

1. Customer satisfaction through early and continuous software delivery

2. Accommodate changing requirements throughout the development process
3. Attention to technical detail and design enhances agility
4. Simplicity
5. Working software is the primary measure of progress
6. Agile processes to support a consistent development pace
7. Enable face-to-face interactions
8. Support, trust, and motivate the people involved
9. Frequent delivery of working software
10. Collaboration between the business stakeholders and developers throughout the project

Because of its addictiveness, Agile methodology is commonly used to deliver more complex projects It uses six main deliverables to track progress and create the product which are the product vision statement, product road map, product backlog, release plan, Sprint backlog, and increment With these features, it establishes itself as a methodology that emphasizes collaboration, flexibility, continuous improvement, and high-quality results



Fig: 3.1 Agile Method

This methodology works in a very simple way by breaking down the Project into smaller parts and continuously delivering them in a cycle called iterations.

3.4 Reason For Using Agile Methodology

Software project management is a discipline where we may need to coordinate many various projects of different scope and deal with different customers who need different results Naturally, we will not be able to use the same management methods for delivering a basic

application developed by a team of four and a complex software solution which needs professionals from various departments Besides, most customers will require us to yield results along the project, so that they can see what kind of product they will get at the end To be a successful project manager, we need to master different software delivery methodologies Basically, there are two of them widely used in software development: Waterfall and Agile Agile Manifesto was put together back in 2001 by a group of developers utilizing different delivery methods Eleven years after, a study by the Standish group was conducted showing that Agile methods yield successful projects three times more often compared to the traditional Waterfall projects The gist of Agile trade is to continually develop shippable software pieces to our client Along the way, our client monitors progress, gives feedback and supervises direction of your work Unlike Waterfall methodology which takes a rigid, sequential structure, Agile methods are more flexible and involve constant cooperation with the client.

3.5 Advantages Of Agile Method

Agile methodologies offer numerous advantages that have made them increasingly popular in software development and beyond. Here are some key advantages of Agile methods:

1. Flexibility and Adaptability: Agile methodologies prioritize flexibility and adaptability, allowing teams to respond quickly to changing requirements, priorities, and market conditions. This iterative approach enables incremental development and frequent feedback loops, ensuring that the final product meets evolving needs effectively.

2. Customer Satisfaction: Agile methodologies emphasize close collaboration with customers or product owners throughout the development process. By involving stakeholders in regular reviews and demonstrations, Agile teams can ensure that the product aligns closely with customer expectations, resulting in higher satisfaction and better outcomes.

3. Early and Continuous Delivery: Agile methodologies promote early and continuous delivery of working software in short iterations, typically ranging from two weeks to a month. This enables stakeholders to see tangible progress and provide feedback early in the development cycle, reducing the risk of costly rework and ensuring that the product meets user needs.

4. Improved Quality: Agile practices such as continuous testing, frequent integration, and regular code reviews help maintain high standards of quality throughout the development process. By identifying and addressing issues early, Agile teams can deliver more reliable, robust, and bug-free software to customers.

5. Increased Transparency: Agile methodologies promote transparency and visibility into the development process through practices such as daily stand-up meetings, sprint reviews, and burndown charts. This transparency fosters trust among team members and stakeholders, enabling better decision-making and collaboration.

6. Empowered Teams: Agile methodologies empower cross-functional teams to self-organize and make decisions collaboratively. By giving team members autonomy and ownership over

their work, Agile fosters a sense of responsibility, creativity, and commitment, leading to higher levels of engagement and productivity.

7. Reduced Risk: Agile methodologies help mitigate project risks by breaking down complex projects into smaller, more manageable increments. By delivering value iteratively and regularly, Agile teams can identify and address risks early, reducing the likelihood of project failure or costly delays.

8. Faster Time-to-Market: Agile methodologies enable faster time-to-market by focusing on delivering incremental value to customers in short cycles. By prioritizing the most valuable features and releasing them early, Agile teams can gain a competitive edge and respond quickly to market demands, maximizing business opportunities.

9. Continuous Improvement: Agile methodologies promote a culture of continuous improvement through practices such as retrospectives, feedback loops, and adaptive planning. By reflecting on their experiences and learning from both successes and failures, Agile teams can identify areas for improvement and implement changes to enhance their processes and practices over time.

10. Cost-Efficiency: Agile methodologies can lead to cost savings by reducing the risk of rework, optimizing resource allocation, and delivering value incrementally. By focusing on delivering the most valuable features first and adapting to changing priorities, Agile teams can maximize the return on investment and minimize waste in the development process.

3.6 Disadvantages Of Agile

While Agile methodologies offer many advantages in software development, they also come with certain disadvantages that may impact projects under certain circumstances. Here are some common disadvantages of Agile:

1. Uncertainty in Requirements: Agile methodologies prioritize flexibility and adaptability, which means that requirements can evolve throughout the project lifecycle. However, this can also lead to uncertainty and instability if requirements are not well-defined or if stakeholders frequently change their needs and priorities.

2. Dependency on Customer Availability: Agile methodologies rely heavily on frequent collaboration and feedback from customers or product owners. If the customer or key stakeholders are not readily available or responsive, it can lead to delays in decision-making and impediments to progress.

3. Overemphasis on Short-Term Goals: Agile frameworks often focus on delivering working software in short iterations, typically ranging from two weeks to a month. While this allows for rapid delivery of incremental improvements, it may sometimes lead to a lack of attention to long-term strategic goals or architectural considerations.

4. Resource Intensive: Agile methodologies require active participation and collaboration from cross-functional teams throughout the project lifecycle. This can be resource-intensive,

particularly for organizations that do not have dedicated teams or struggle to allocate sufficient time and resources to Agile ceremonies and activities.

5. Risk of Scope Creep: Agile methodologies encourage embracing change and responding to customer feedback, which can sometimes result in scope creep. Without careful management, this continuous influx of new requirements can lead to the project scope expanding beyond the original estimates, causing delays and budget overruns.

6. Challenges with Distributed Teams: Agile practices such as daily stand-up meetings, face-to-face collaboration, and co-location may pose challenges for distributed teams or those working across different time zones. Communication barriers and cultural differences can hinder effective collaboration and coordination among team members.

7. Documentation Challenges: Agile methodologies prioritize working software over comprehensive documentation. While this promotes flexibility and responsiveness, it can sometimes result in inadequate documentation, making it challenging for new team members to onboard or for stakeholders to understand project history and decisions.

8. Difficulty in Estimation: Agile projects often rely on relative estimation techniques such as story points or t-shirt sizing, which can be challenging for teams, particularly those new to Agile. Estimating accurately without detailed requirements or historical data can lead to inaccuracies in project planning and resource allocation.

9. Limited Predictability: Agile methodologies embrace change and adaptability, which can make it difficult to predict project timelines, milestones, and outcomes with a high degree of certainty. This lack of predictability may pose challenges for stakeholders who require fixed deadlines or budgetary constraints.

10. Lack of Formal Governance: Agile frameworks promote self-organizing, empowered teams, which may lead to a lack of formal governance and oversight in some cases. Without adequate governance mechanisms in place, there may be risks related to quality assurance, compliance, and alignment with organizational objectives.

CHAPTER 4

SYSTEM DESIGN

4.1 Requirement Specification

A requirement specification document for an online voting system outlines the functional and non-functional requirements that the system must meet to ensure its successful design, development, and implementation.

1. Introduction: Provides an overview of the document, its purpose, and the scope of the online voting system project.

2. Stakeholder Identification: Identifies the stakeholders involved in the online voting system, including voters, electoral authorities, system administrators, and other relevant parties.

3. Functional Requirements:

Voter Registration: Describes the process by which eligible voters can register on the online voting platform, including verification of voter eligibility and creation of secure user accounts.

Authentication and Authorization: Specifies the methods used to authenticate voters' identities securely and grant them access to the voting platform.

Ballot Presentation: Defines how the ballot will be presented to voters, including candidate profiles, ballot measures, and any other voting options.

Voting Process: Outlines the steps involved in the voting process, including selecting candidates or options, reviewing and confirming selections, and submitting votes electronically.

Security Measures: Specifies the security features and protocols implemented to protect the integrity, confidentiality, and authenticity of the voting process, including encryption, multi-factor authentication, and audit trails.

Vote Tabulation and Result Reporting: Describes how votes will be tabulated, aggregated, and reported in real-time to provide immediate visibility into the outcome of the election.

Accessibility: Ensures that the online voting system is accessible to all voters, including those with disabilities, by complying with relevant accessibility standards and guidelines.

Integration with Electoral Systems: Specifies any integration requirements with existing electoral systems or databases to ensure accurate voter registration and authentication.

Scalability and Performance: Defines the system's capacity to handle a large volume of concurrent users and transactions, ensuring optimal performance during peak voting periods.

Auditability and Compliance: Outlines requirements for logging and auditing voting activities to ensure compliance with legal and regulatory standards, as well as for conducting post-election audits if necessary.

4. Non-functional Requirements:

Usability: Specifies usability requirements to ensure that the online voting system is intuitive, easy to navigate, and accessible to users with varying levels of technical expertise.

Reliability: Defines requirements for system availability, reliability, and fault tolerance to minimize downtime and ensure continuous operation during elections.

Security: Specifies security requirements to protect against cyber threats, unauthorized access, tampering, and fraud, including data encryption, intrusion detection, and regular security audits.

Performance: Defines performance requirements, such as response times, throughput, and system scalability, to ensure that the online voting system can handle peak loads without degradation in performance.

Compatibility: Specifies compatibility requirements with different web browsers, operating systems, and devices to ensure broad accessibility and usability across diverse user environments.

Compliance: Ensures that the online voting system complies with relevant legal, regulatory, and industry standards, including data privacy laws, election regulations, and security best practices.

Data Backup and Recovery: Defines requirements for data backup, disaster recovery, and contingency planning to mitigate the risk of data loss or system failures and ensure the integrity of voting data.

5. Assumptions and Constraints: Documents any assumptions made and constraints imposed during the development of the online voting system, such as budget limitations, technological dependencies, or regulatory constraints.

6. Dependencies: Identifies any external dependencies or third-party systems that the online voting system relies on for its operation, such as authentication services or voter registration databases.

7. Appendices: Includes any additional supporting documentation, reference materials, or technical specifications relevant to the online voting system project.

4.2 System Flow Chart

User Flow

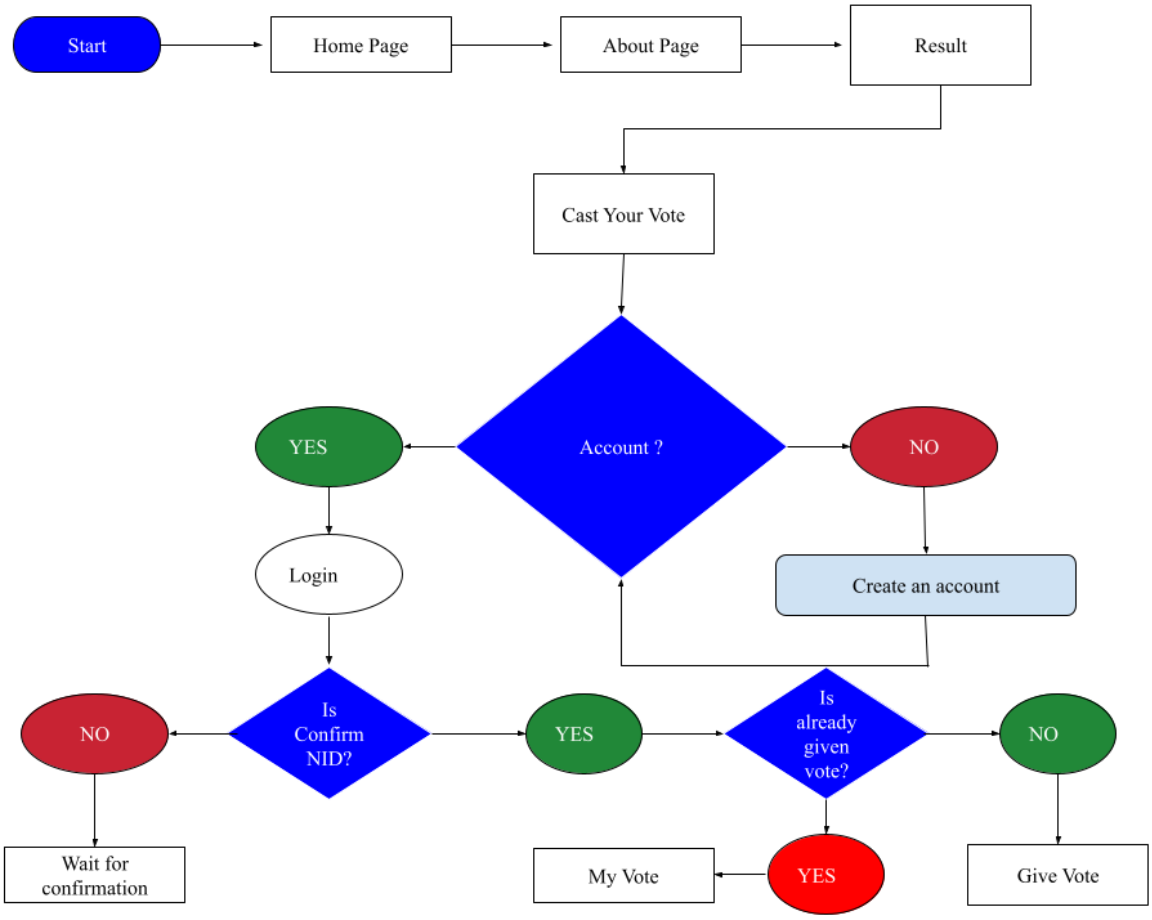


Fig: 4.1 User Flow

Candidate Flow of application

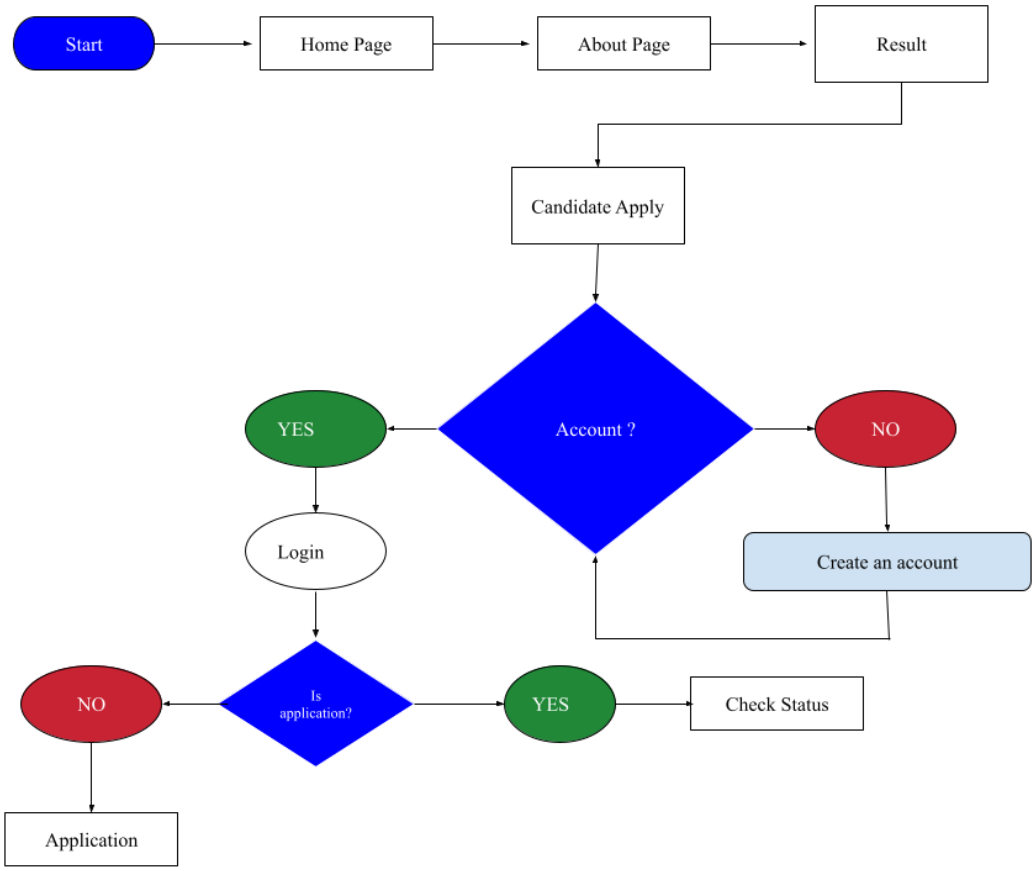


Fig: 4.2 Candidate Flow

Admin Flow

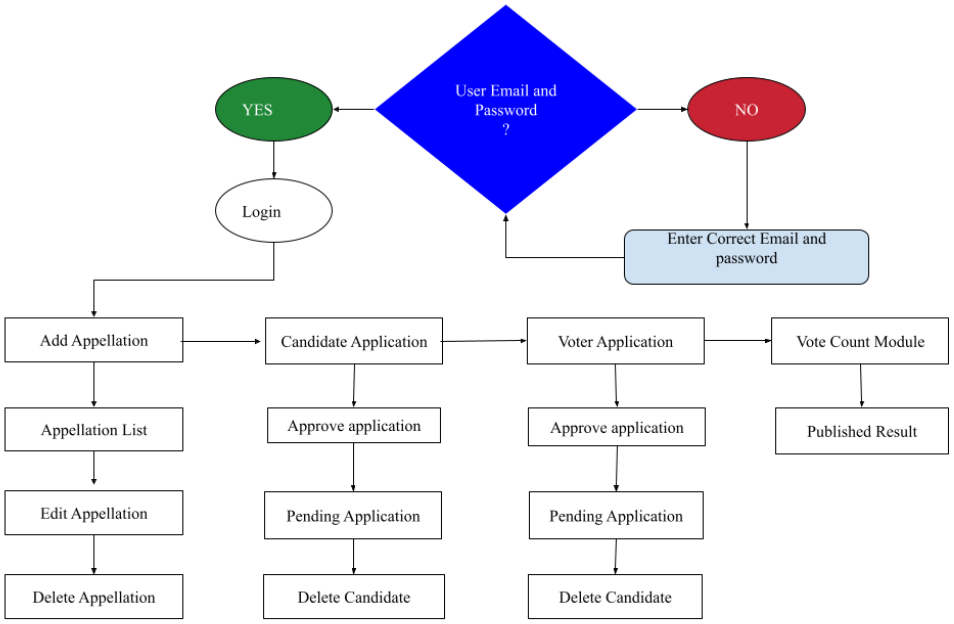


Fig: 4.3 Admin Flow

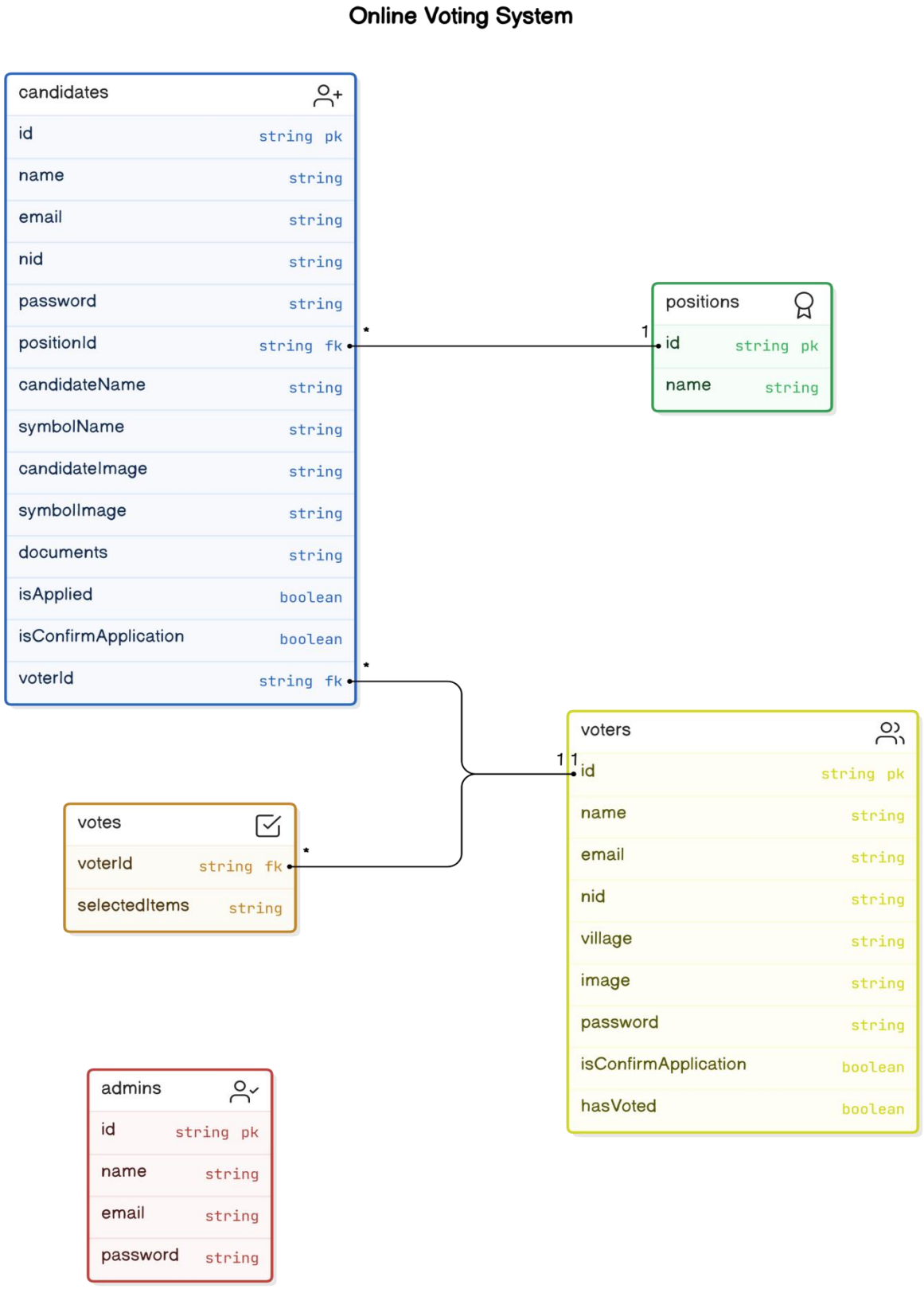


Fig: 4.4 ER Diagram

CHAPTER 5

IMPLEMENTATION AND OUTCOME

5.1 Tools and Technology Used for Development

In a MERN (MongoDB, Express.js, React, Node.js) stack project, various tools and technologies are commonly used for development, testing, and deployment. Here's a list of tools often employed in each component of the MERN stack:

MongoDB (Database):



Fig: 5.1 MongoDB

MongoDB is a popular open-source NoSQL database management system that falls under the category of document-oriented databases. It is designed to store, manage, and retrieve data in a flexible, scalable, and high-performance manner. MongoDB is known for its ability to handle large volumes of unstructured or semi-structured data, making it particularly well-suited for applications with evolving and dynamic schemas.

Key features of MongoDB include:

Document-Oriented: MongoDB stores data in flexible, JSON-like documents known as BSON (Binary JSON). Each document can have a different structure, allowing for easy representation of complex data.

Scalability: MongoDB can scale horizontally by distributing data across multiple servers and can handle large amounts of data and high traffic.

Dynamic Schema: Unlike traditional relational databases, MongoDB does not require a predefined schema. Fields can be added to documents on the fly, making it easy to adapt to changing data requirements.

Query Language: MongoDB supports a rich set of query operators and indexing, allowing for efficient querying and retrieval of data. It also supports geospatial queries for location-based data.

Aggregation Framework: MongoDB provides a powerful aggregation framework that allows for complex data transformation and analysis operations.

High Performance: MongoDB is designed to deliver high performance by using internal memory for storing the working set of data, resulting in faster data access times.

Replication: MongoDB supports automatic data replication, providing high availability and fault tolerance. In the event of a server failure, a replica set can promote a secondary node to the primary role.

Sharding: MongoDB can horizontally scale data across multiple servers using sharding. This allows for distributing large datasets across multiple machines to improve performance.

Support for Multiple Data Types: MongoDB supports various data types, including strings, numbers, arrays, and binary data, making it versatile for diverse data storage needs.

We choose MongoDB for projects because of flexibility, scalability, and speed are crucial, such as in web applications, content management systems, real-time analytics, and more. The MongoDB community and ecosystem have grown significantly, and it is widely used in the development of modern, data-intensive applications.

Express.js (Back-end):



Fig: 5.2 NODE + EXPRESS

Express.js is a minimalist and flexible web application framework for Node.js, designed to simplify the process of building robust and scalable back-end applications. It provides a set of features and tools for developing web and mobile applications, APIs (Application Programming Interfaces), and server-side logic. Express.js is widely used in combination with other technologies, forming a part of the MERN (MongoDB, Express.js, React, Node.js) and MEAN (MongoDB, Express.js, Angular, Node.js) stacks.

Key features and characteristics of Express.js include:

Routing: Express.js simplifies the process of defining routes for handling HTTP requests. It allows developers to map different HTTP methods (GET, POST, PUT, DELETE, etc.) to specific functions or handlers.

Middleware: Middleware functions are a central concept in Express.js. They can be used to execute code during the request-response cycle. Middleware functions can handle tasks such as authentication, logging, data parsing, and more.

HTTP Utility Methods: Express.js provides a set of utility methods for handling HTTP requests and responses, making it easier to work with common tasks like sending JSON responses, handling redirects, and setting HTTP headers.

Template Engines: While not included by default, Express.js can be extended with template engines like EJS or Pug to dynamically generate HTML on the server side.

Static File Serving: Express.js allows the serving of static files, such as images, CSS, and JavaScript, directly from a specified directory. This simplifies the setup of static assets for web applications.

Error Handling: Express.js provides mechanisms for handling errors at the application level. Developers can define custom error-handling middleware to manage errors in a centralized manner.

RESTful API Development: Express.js is commonly used for building RESTful APIs. It makes it easy to define routes and handlers for creating, reading, updating, and deleting resources.

Integration with Other Middleware and Libraries: Express.js can be extended using a wide range of middleware and third-party libraries, allowing developers to add functionality seamlessly. Examples include Passport for authentication and Morgan for request logging.

Security Features: While security is a broad concern that extends beyond the framework itself, Express.js provides features such as middleware for handling Cross-Origin Resource Sharing (CORS) and Helmet for setting HTTP security headers.

Community and Ecosystem: Express.js has a large and active community, and it is widely used in industry. The ecosystem includes a variety of plugins, middleware, and tools that enhance its capabilities.

Express.js is well-suited for building web applications and APIs due to its simplicity, flexibility, and ability to integrate seamlessly with other tools and libraries in the Node.js ecosystem. It's particularly favored for projects where a lightweight and unopinionated framework is desirable, allowing developers to have more control over the architecture of their applications.

React (Front-end):

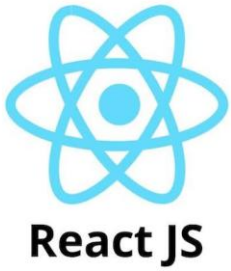


Fig: 5.3 React Js

React, developed and maintained by Facebook, is a popular open-source JavaScript library for building user interfaces. It is widely used for developing interactive and dynamic front-end applications. React follows a component-based architecture, where the user interface is divided into reusable components, making it easier to manage and maintain complex UIs. React is a key component in the MERN (MongoDB, Express.js, React, Node.js) and MEAN (MongoDB, Express.js, Angular, Node.js) stacks.

Here are some key features and characteristics of React:
Component-Based Architecture: React applications are built using components, which are self-contained, reusable building blocks.
Components can be nested, and their state and behavior can be managed independently.
Declarative Syntax: React uses a declarative syntax, making it easy to understand and predict how the UI will behave based on the current state.
Developers describe what the UI should look like, and React takes care of updating the DOM efficiently.
Virtual DOM: React utilizes a virtual DOM to optimize rendering performance.
Changes are first made to a virtual representation of the DOM, and React calculates the most efficient way to update the actual DOM, minimizing unnecessary manipulations.

JSX (JavaScript XML): JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within their JavaScript files.
JSX is compiled into JavaScript and enables a more readable and expressive way to define React components.
Unidirectional Data Flow: React follows a unidirectional data flow, meaning data flows in one direction from parent components to child components.
This helps maintain a predictable state and makes it easier to debug and understand the application's behavior.
React Hooks: Introduced in React 16.8, hooks are functions that allow functional components to have state and lifecycle features.
useState and useEffect are common hooks used for managing state and handling side effects, respectively.
Reusable Components: React promotes the creation of reusable components, improving code modularity and maintainability.
Components can be easily shared across different parts of an application or even between different projects.
React Router: React Router is a popular library for handling navigation in React applications.
It enables the creation of single-page applications (SPAs) by managing the URL and rendering components based on the current route.
Ecosystem and Community: React has a large and active community, resulting in a rich ecosystem of libraries, tools, and extensions.
The community contributes to the development of best practices, patterns, and third-party libraries.
State Management: While React provides local state management, for more complex applications, state management libraries like Redux or Context API are often used to manage the state globally.
React is widely used in the development of modern web applications, including single-page applications, progressive web apps, and mobile applications. Its focus on simplicity, performance, and reusability has contributed to its widespread adoption in the industry.

Node.js (Server-side):



Fig: 5.4 Node Js

Node.js is an open-source, server-side JavaScript runtime environment that allows developers to run JavaScript code outside of a web browser. It is built on the V8 JavaScript runtime engine, which is the same engine that powers Google Chrome. Node.js is designed to be lightweight, efficient, and scalable, making it well-suited for building server-side applications, APIs (Application Programming Interfaces), and networked software.

Key features and characteristics of Node.js include:

Non-blocking I/O: Node.js is known for its non-blocking, asynchronous I/O model, which allows it to handle many concurrent connections efficiently.

This is particularly beneficial for applications that involve handling a large number of simultaneous connections, such as real-time applications.

Event-Driven Architecture: Node.js is event-driven, meaning it uses an event loop to handle events and execute callbacks when an event occurs.

This architecture enables the creation of highly scalable and responsive applications.

Single-threaded, Event Loop: Node.js operates on a single-threaded event loop, but it can still handle concurrent requests by using asynchronous, non-blocking operations.

The event loop efficiently manages I/O operations and executes callbacks when events are triggered.

NPM (Node Package Manager): NPM is the default package manager for Node.js, providing a vast ecosystem of reusable packages and modules.

Developers can easily manage dependencies, share code, and leverage a wide range of third-party libraries.

Cross-Platform: Node.js is cross-platform and can run on various operating systems, including Windows, macOS, and Linux.

This makes it convenient for developers to write and deploy code across different environments.

V8 JavaScript Engine: Node.js uses the V8 JavaScript engine, developed by Google, to execute JavaScript code.

V8 compiles JavaScript into machine code, resulting in high-performance execution.

Scalability: Node.js is designed to be scalable, making it suitable for building applications that need to handle a large number of concurrent connections.

It is often used in scenarios where real-time communication and scalability are crucial.

Community and Ecosystem: Node.js has a vibrant and active community, contributing to its extensive ecosystem of libraries, frameworks, and tools.

The community plays a significant role in shaping best practices and fostering innovation.

Support for Web Servers and Frameworks: While Node.js itself provides a basic HTTP module for building web servers, developers often use

frameworks to streamline the development process.

Popular Node.js frameworks include Express.js, Nest.js, and Koa.js, which offer routing, middleware, and other features for building web applications.

Microservices Architecture: Node.js is well-suited for a microservices architecture, where applications are built as a collection of small, independently deployable services.

Its non-blocking nature and event-driven architecture make it conducive to handling microservices communication efficiently.

Real-time Applications: Node.js is often chosen for real-time applications such as chat applications, online gaming, and collaborative tools.

Its ability to handle concurrent connections and perform well in real-time scenarios makes it a popular choice for these applications.

Security: While Node.js itself is secure, developers need to follow best practices to ensure the security of their applications.

Security considerations include protecting against common vulnerabilities, secure coding practices, and implementing proper authentication and authorization mechanisms.

Node.js is widely adopted in the industry and has become a popular choice for building scalable and performant server-side applications. Its ability to handle concurrent connections efficiently, along with a vast ecosystem and active community, makes it a versatile and powerful technology for a variety of use cases.

Testing Tools:

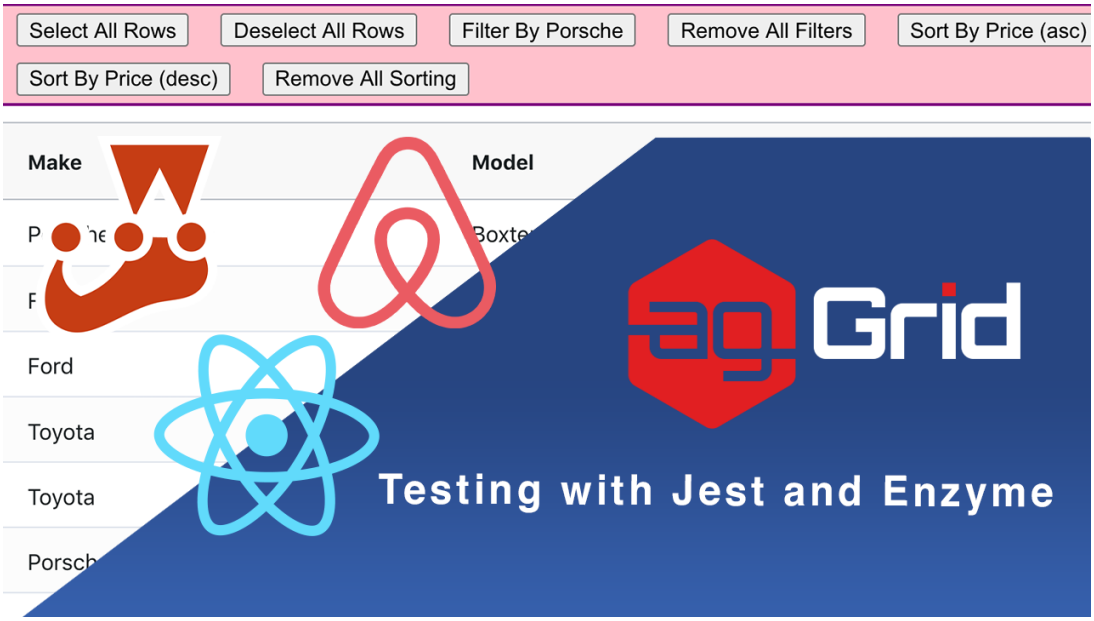


Fig: 5.5 Testing Framework

Jest:

- **Type:** Testing Framework
- **Purpose:** Jest is a widely used JavaScript testing framework developed by Facebook.
- It is primarily designed for unit testing, but it also supports integration testing.
- **Features:** Provides a built-in assertion library for making test assertions. Supports snapshot testing for easily tracking changes in UI components. Offers parallel test execution for faster results. Integrates seamlessly with React applications.

Enzyme:

- **Type:** Testing Utility
- **Purpose:** Enzyme is a JavaScript testing utility developed by Airbnb for testing React components. It provides a set of helper functions to facilitate component testing.
- **Features:** Allows shallow rendering of components to isolate and test individual components without rendering child components. Provides APIs for interacting with and inspecting rendered components. Supports a variety of rendering methods, including shallow, mount, and render. Works well with Jest and other testing libraries.

Supertest:

- **Type:** Testing Library

- **Purpose:** Supertest is a library for testing HTTP assertions with Node.js HTTP servers, commonly used with Express.js. It enables the testing of APIs by making HTTP requests and asserting the responses.
- **Features:** Supports a simple and expressive API for making HTTP requests. Allows testing of various HTTP methods, headers, and request payloads. Integrates well with assertion libraries like Jest for validating responses. Can be used for both unit testing and end-to-end testing of API endpoints.

These testing tools, when used in conjunction, provide a comprehensive approach to testing JavaScript applications, especially those built with React and Node.js. Jest handles the overall testing framework and assertions, Enzyme focuses on testing React components, and Supertest facilitates testing of HTTP assertions in the context of Express.js or other Node. js-based web servers. Integrating these tools into your testing strategy helps ensure the reliability and correctness of your applications.

Collaboration and Communication:

Effective collaboration and communication tools are essential for teams to work cohesively, share information, and coordinate efforts. Here are the tools that we use to collaborate and communicate for our project:

Slack:

- **Type:** Team Communication
- **Features:**
    - Real-time messaging for teams and channels.
    - Integration with various apps and services.
    - File sharing, direct messaging, and threaded conversations.
    - Channels for organizing discussions based on topics or projects.

Trello:

- **Type:** Project Management
- **Features:**
    - Kanban-style boards for project organization.
    - Cards and lists for task management.
    - Collaboration through comments and file attachments.
    - Integration with other productivity tools.

Zoom:

- **Type:** Video Conferencing
- **Features:**
    - Video and audio conferencing.
    - Screen sharing and collaboration during meetings.
    - Breakout rooms for smaller group discussions.
    - Webinars and virtual events.

Google Workspace (formerly G Suite):

- **Type:** Productivity Suite
- **Features:**
    - Gmail for email communication.
    - Google Drive for file storage and collaboration.
    - Google Docs, Sheets, and Slides for document collaboration.
    - Google Meet for video conferencing.

Jira:

- **Type:** Issue and Project Tracking
- **Features:**
    - Advanced project management and issue tracking.
    - Customizable workflows and agile boards.
    - Integration with development tools and version control systems.
    - Reporting and analytics for project insights.

GitHub:

- **Type:** Version Control and Collaboration
- **Features:**
    - Git-based version control for code repositories.
    - Collaboration through pull requests and code reviews.
    - Issues and project boards for task tracking.
    - Git actions for automated workflows.

We used these tools for developing a knowledge fight web project. The specific tools and technologies used will depend on the project requirements and the preferences of the development team. There are different types of modules also needed to install in those software. So, it is very important to take the correct options for development

5.2 Software

The operating system used for the development of this application is Microsoft Windows 10 Professional The IDE that is used for coding is Visual Studio code (community version) The programming language used for the development of this application is Javascript For the structure of this project there used JSX and External CSS

Software	Used For	Version
Microsoft Windows 10 pro	Operating System	10
Visual Studio Code	Developing Software Applications	
Node Js	To Build Scalable And High-Performance Applications	14.15.1
CSS	Styling Web Site	CSS3
Google Chrome	Browser	110.0. 5481.178
NPM	Node	6.14.8
MongoDB	Database	6.0
Git	Manage Version	2.32.0.windows.2
Postman	testing and interacting with APIs	10.10.9
Bootstrap	create responsive, mobile-first web applications	5.0.2

Table 5.1 Software Used For Development

5.3 Hardware

The hardware used for the development of this application is shown in the table below Processor used for this development is an Intel Core i5 11^{th} generation Total of 8GB of memory is used to run the necessary software for the development of this application other hardware components that have been used are shown in the table below:

Hardware	Description
Processor	Intel Core i5 11th Gen Processor
Memory	HP 8GB DDR4 1333 MHz RAM
Motherboard	Intel(4) H81 Express Chipset
Storage	Western Digital 512GB SSD
Display	Dell 155-inch Inspiration 15 laptop display

Table 5.2 Hardware Used For Development

5.4 Database Service

In our project, we use the MongoDB database. We use cloud storage for developing this project. MongoDB is a popular NoSQL document database service that is used for storing and retrieving large volumes of unstructured data. Unlike traditional SQL databases, MongoDB is designed to be flexible and scalable, making it ideal for use in modern web applications. Some of the key features of MongoDB include:

1. Flexible data model: MongoDB uses a document data model that allows developers to store and retrieve data in a way that makes sense for their specific application. This flexible data model makes it easy to add or modify data without having to restructure the entire database. Version control: Version control systems such as Git and Github are essential for managing
2. Scalability: MongoDB is designed to be horizontally scalable, which means it can easily handle large volumes of data and high levels of traffic. This makes it ideal for use in modern web applications that need to handle large amounts of user-generated content.
3. Performance: MongoDB is optimized for performance, with features like automatic sharding and index support that help to ensure fast and efficient query processing.
4. Cloud integration: MongoDB is designed to work seamlessly with cloud-based services like AWS and Azure, making it easy to deploy and manage MongoDB databases in the cloud.
5. Active community: MongoDB has a large and active community of developers, which means there are plenty of resources and support available for developers who are new to the platform.

MongoDB is a powerful and flexible database service that is well-suited for use in the knowledge fight web project. Its scalability, flexibility, and performance make it an ideal choice for storing and managing large volumes of data in modern web applications.

5.5 Implementation

The implementation of a knowledge fight web project involves several stages, including:

1. Planning: In this stage, the project team will define the project scope, objectives, and deliverables. They will also identify the project stakeholders, set a timeline for the project, and allocate resources. Version control: Version control systems such as Git and Github are essential for managing.
2. Design: During the design stage, the project team will create wireframes and mockups of the web application, design the user interface, and define the database schema.
3. Development: The development stage involves writing code and building the web application. The project team will use tools like Visual Studio Code, Node.js, and MongoDB to build the application.
4. Testing: In the testing stage, the project team will test the web application to ensure that it is functioning properly and meets the project requirements. They will use tools

like Postman to test the API endpoints and ensure that the data is being stored and retrieved correctly.

5. Deployment: Once the web application has been tested and approved, it can be deployed to a production environment. The project team will use cloud services like AWS or Azure to deploy the application, ensuring that it is secure and accessible to users.
6. Maintenance: After the web application has been deployed, the project team will continue to maintain it, making updates and improvements as needed. They will monitor the application for bugs and performance issues, and make updates to the code and database schema as necessary.

The implementation of a knowledge fight web project requires careful planning, design, development, testing, deployment, and maintenance. By following a structured approach, the project team can ensure that the web application is high-quality, functional, and meets the needs of its intended audience.

5.6 Public View



Fig: 5.6 Home Page

About Page:



Fig: 5.7 About Page

Result Page

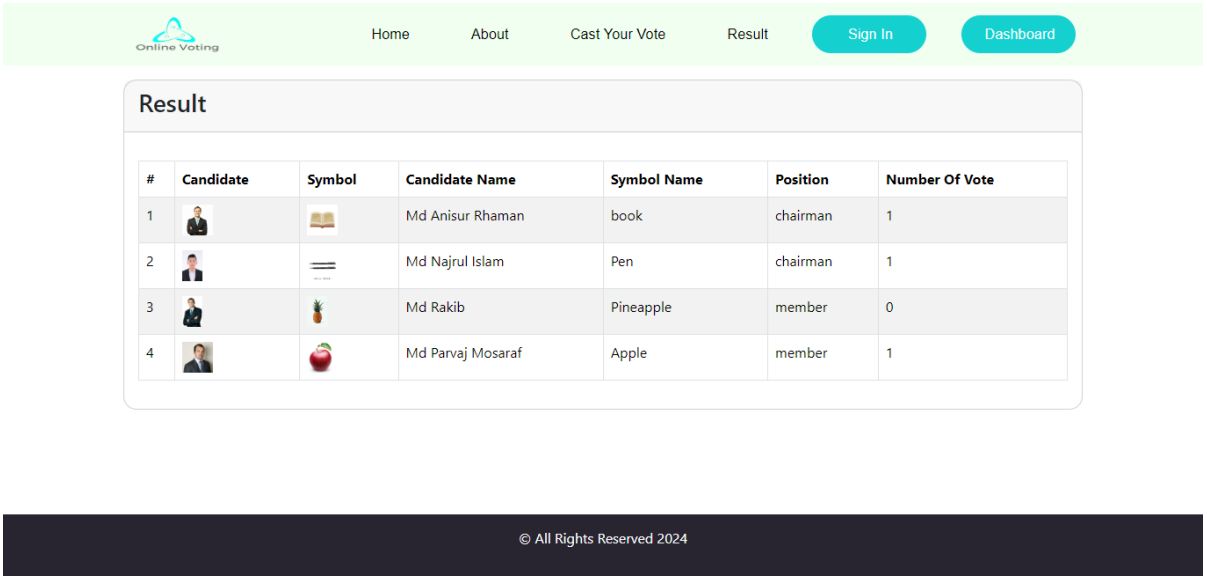


Fig: 5.8 Result Page

Voter Login

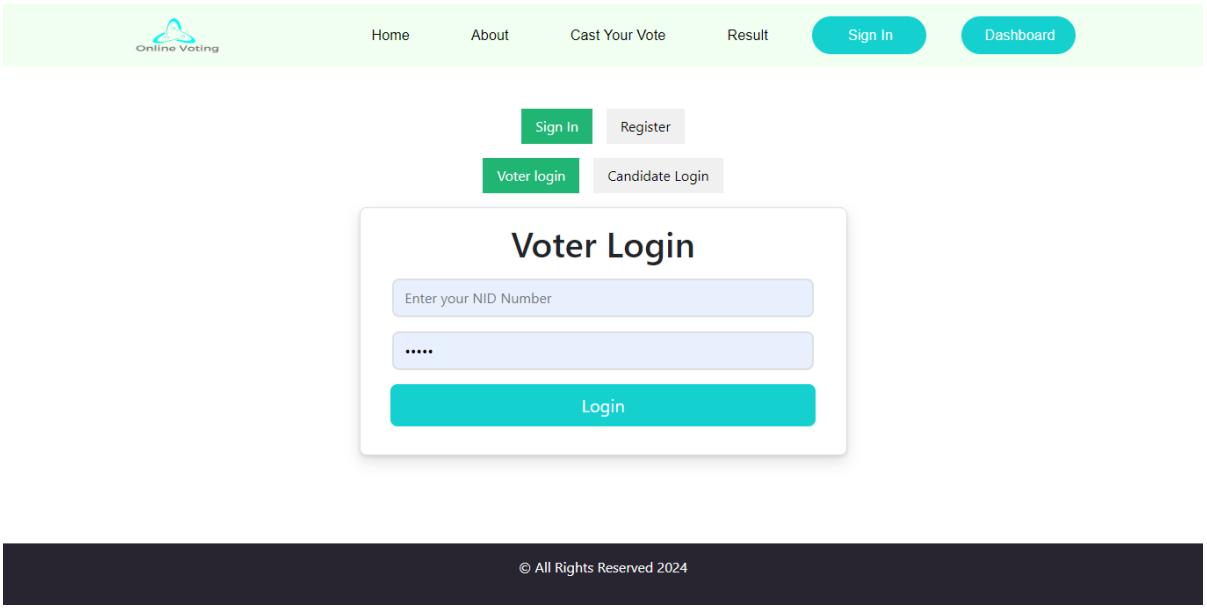


Fig: 5.9 Voter Login

Candidate Login

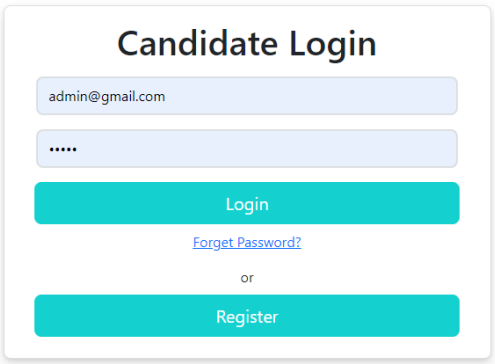


Fig: 5.10 Candidate Login

Voter Register

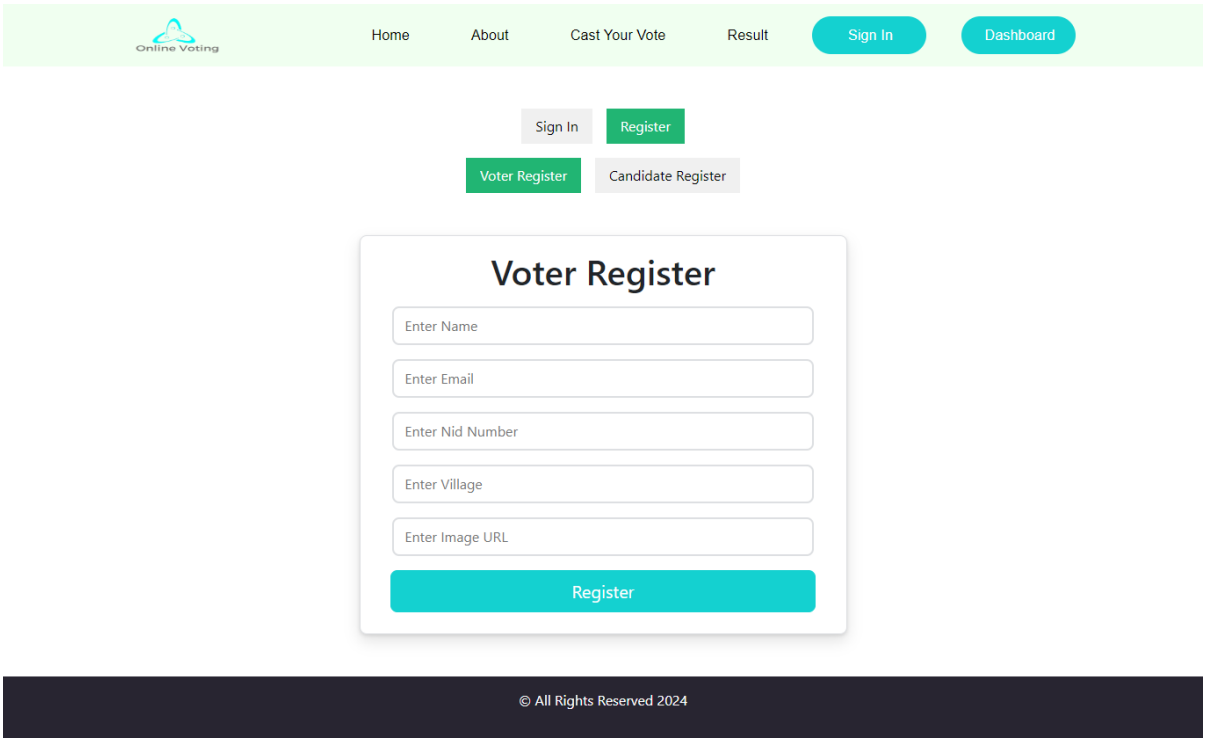


Fig: 5.11 Voter Register

Candidate Register

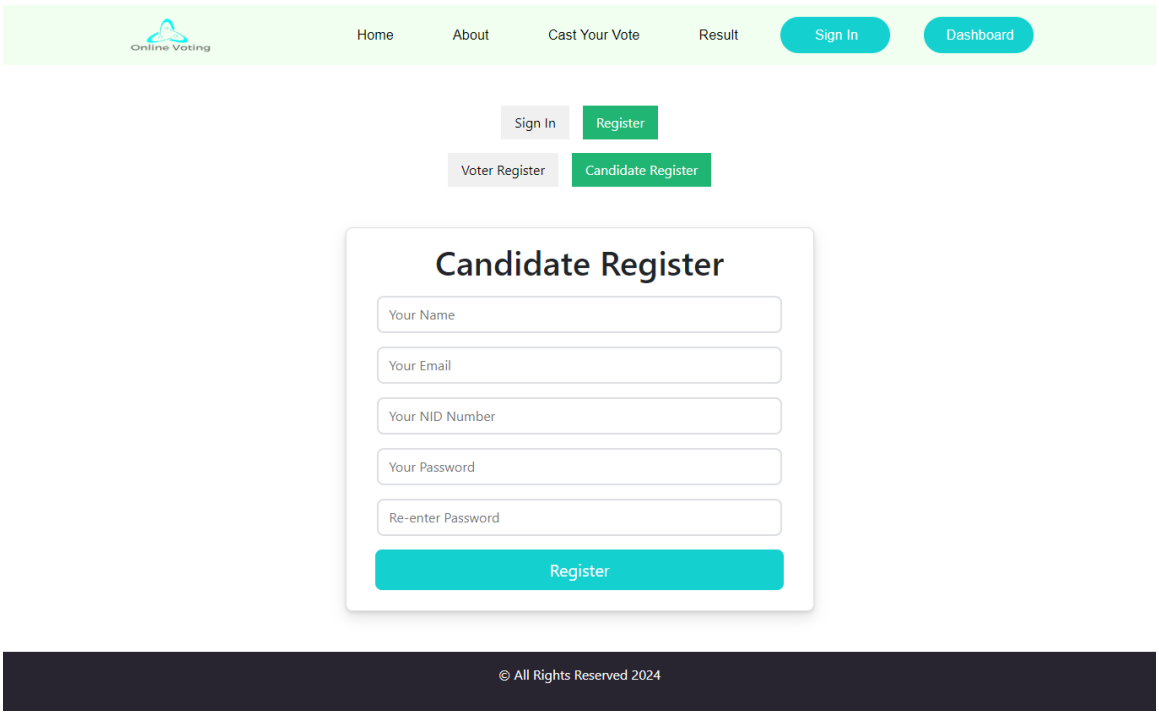


Fig: 5.12 Candidate Register

5.7 Voter View

Give Vote

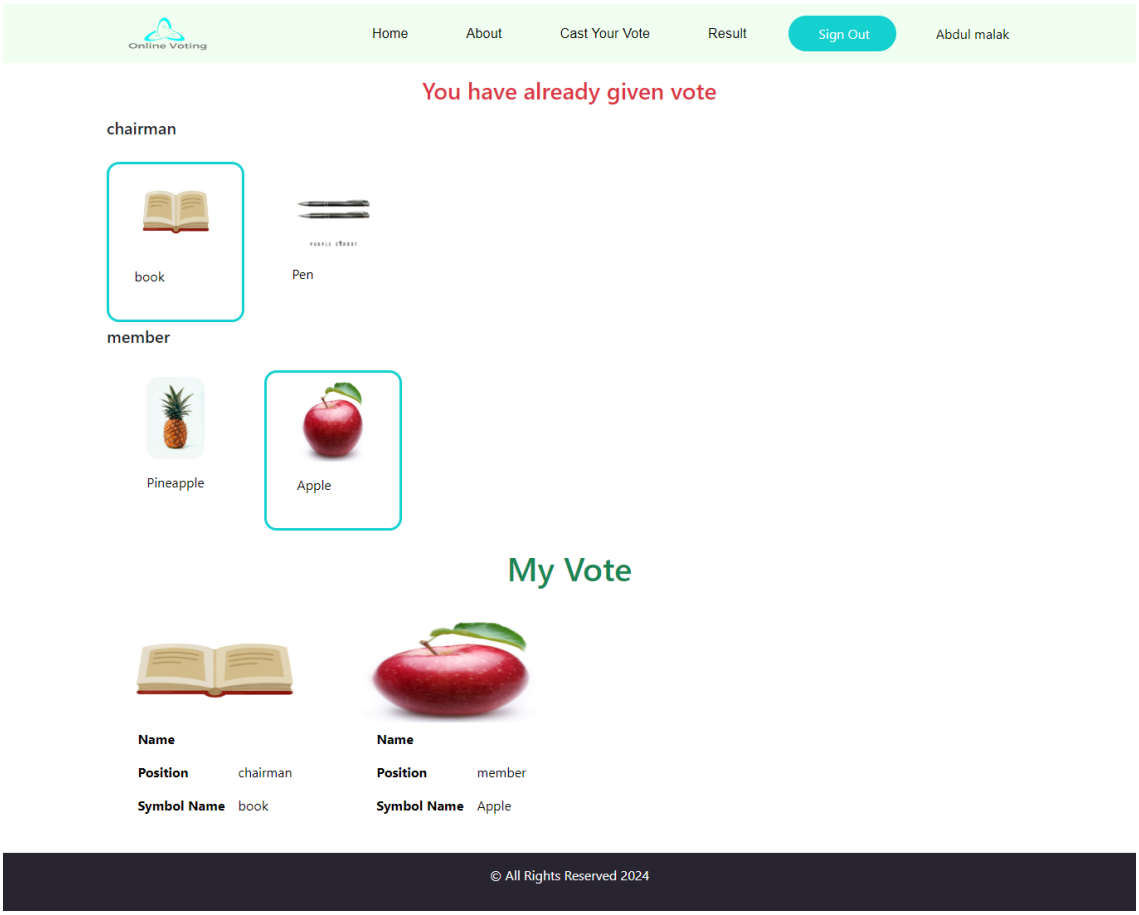


Fig: 5.13 Give Vote

5.8 Admin View

Admin Login

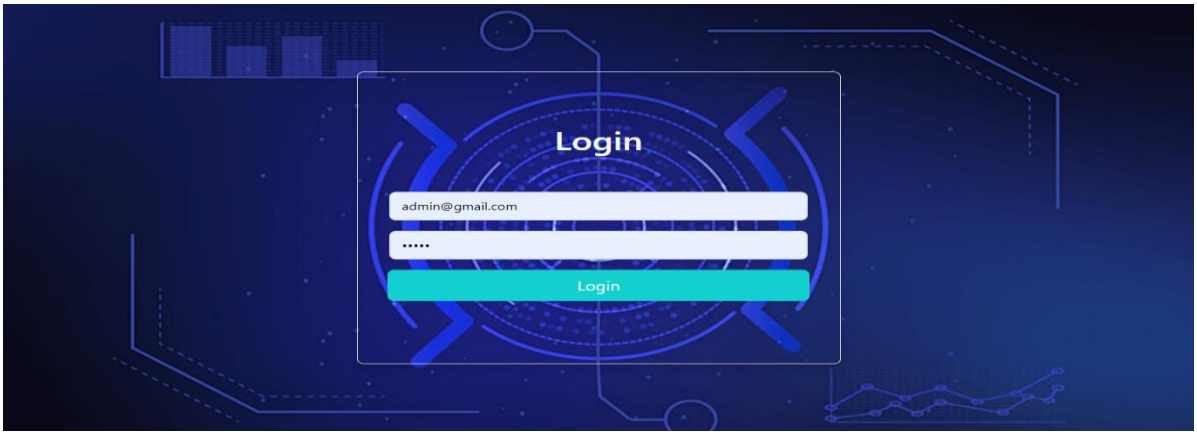


Fig: 5.14 Admin Login

Admin Dashboard

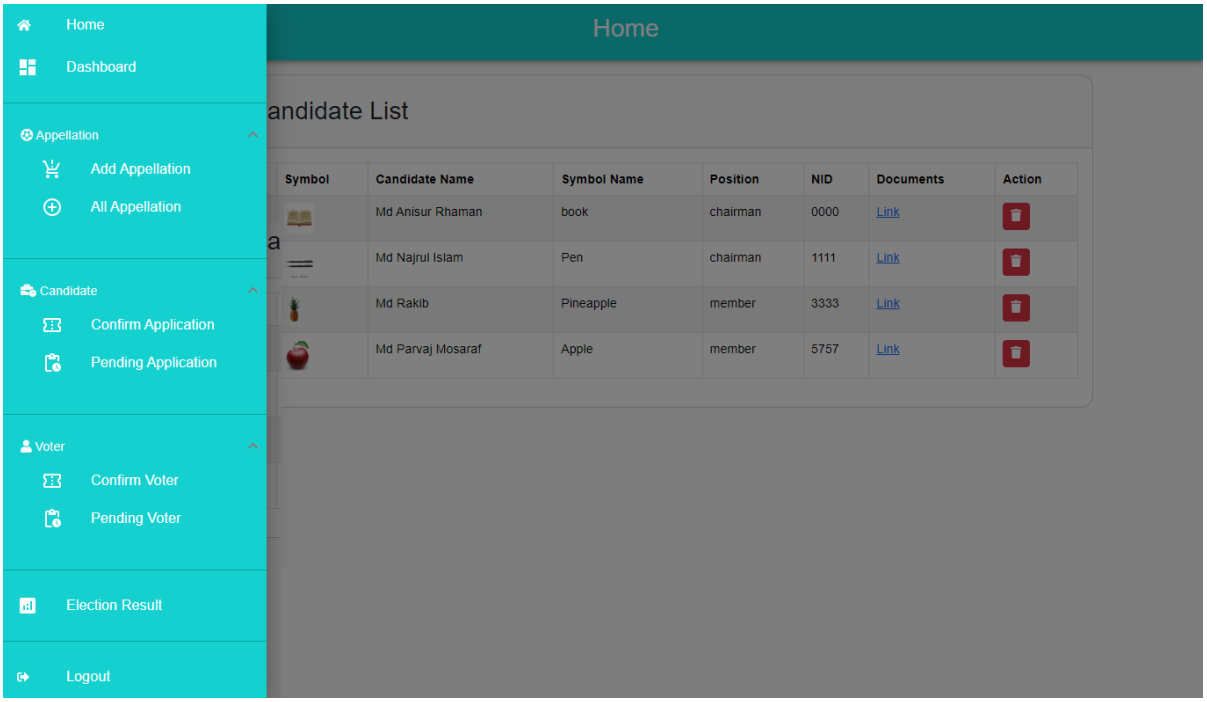


Fig: 5.15 Admin Dashboard

Appellation List

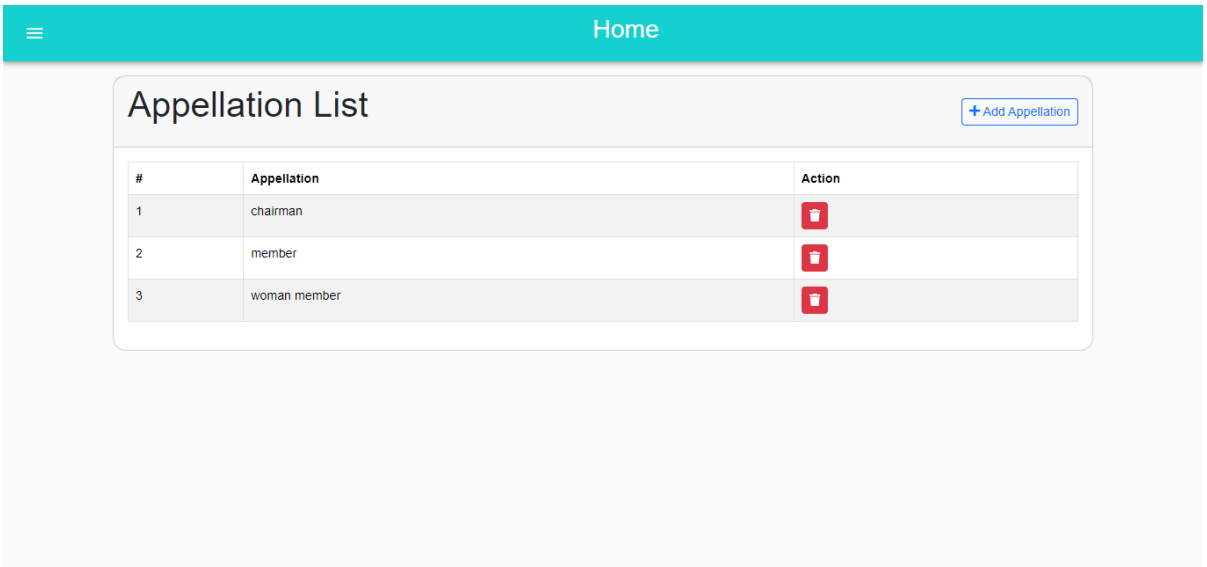


Fig: 5.16 Appellation List

Pending Candidate

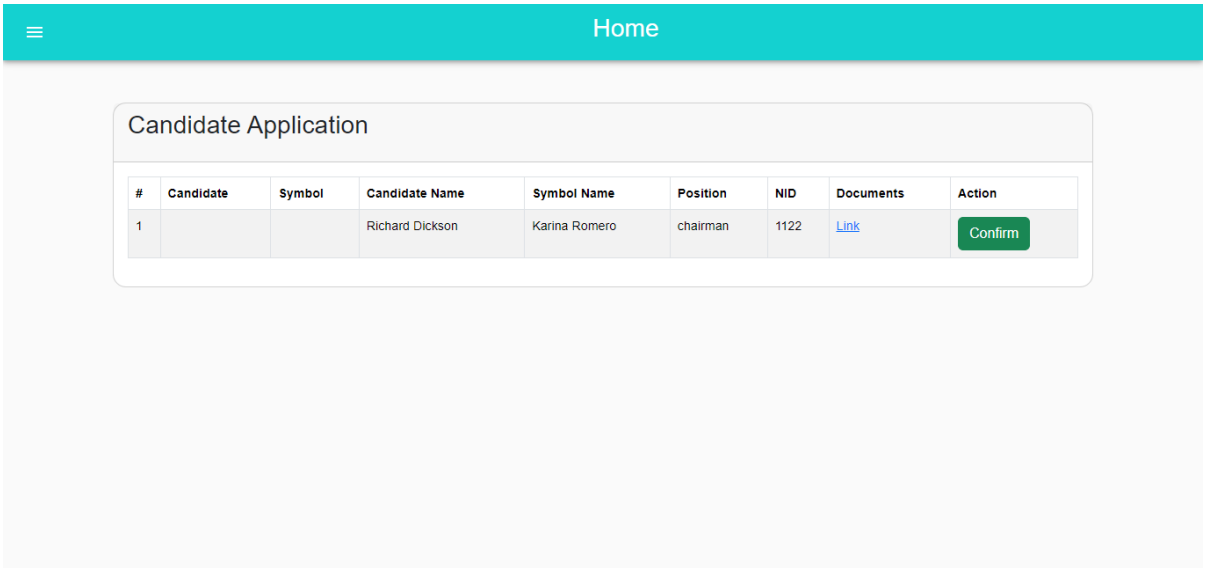


Fig: 5.17 Pending Candidate

Confirm Candidate

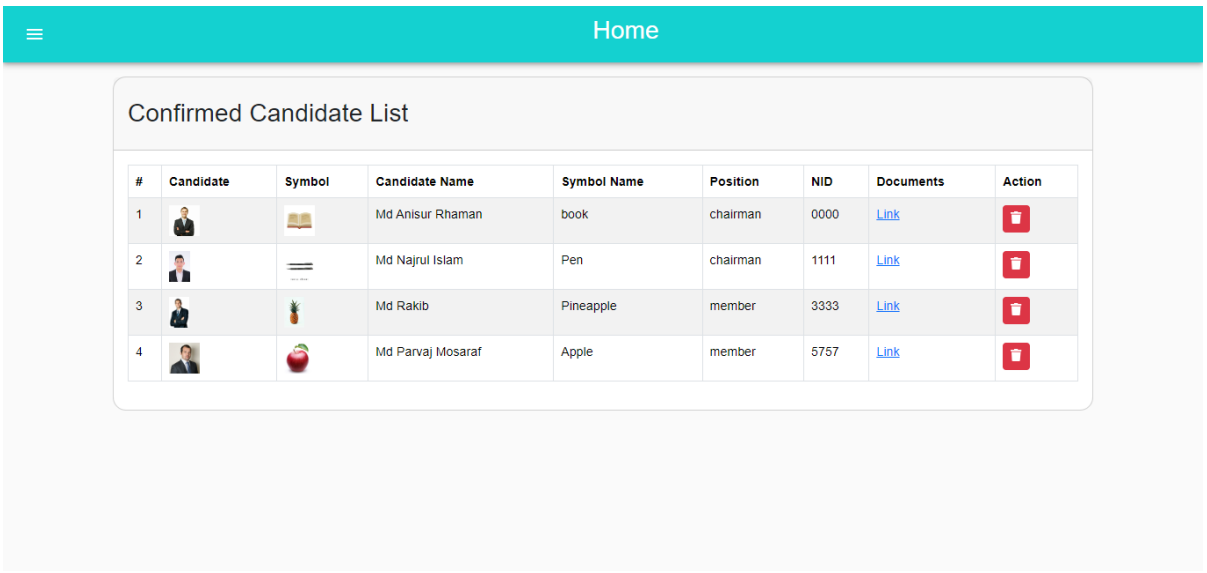


Fig: 5.18 Confirm Candidate

Confirm Voter

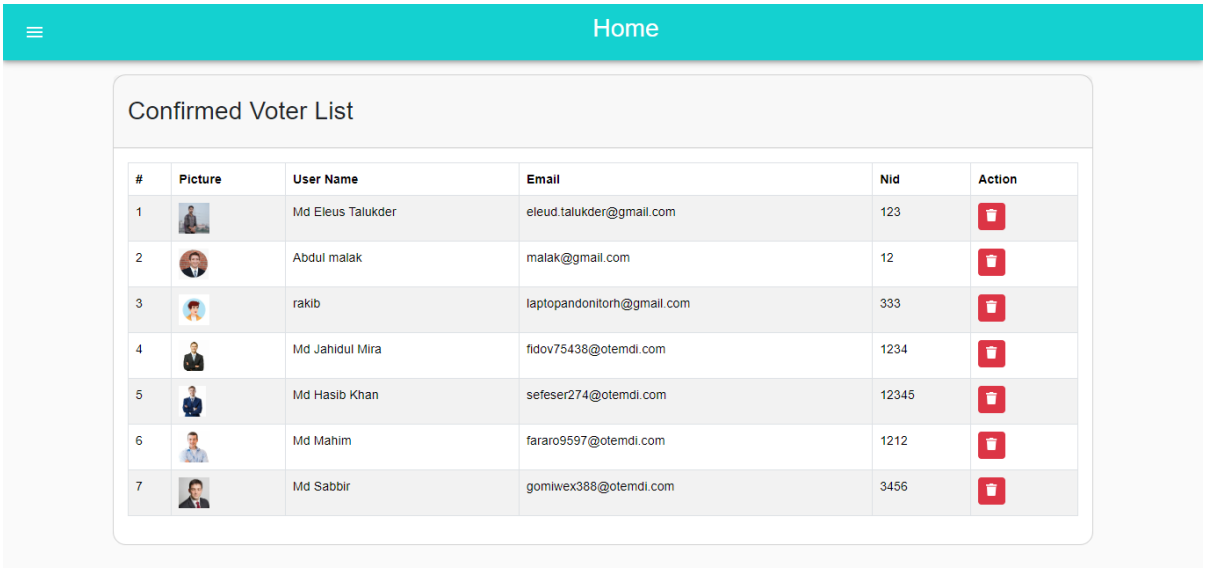


Fig: 5.19 Confirm Voter

Pending Voter

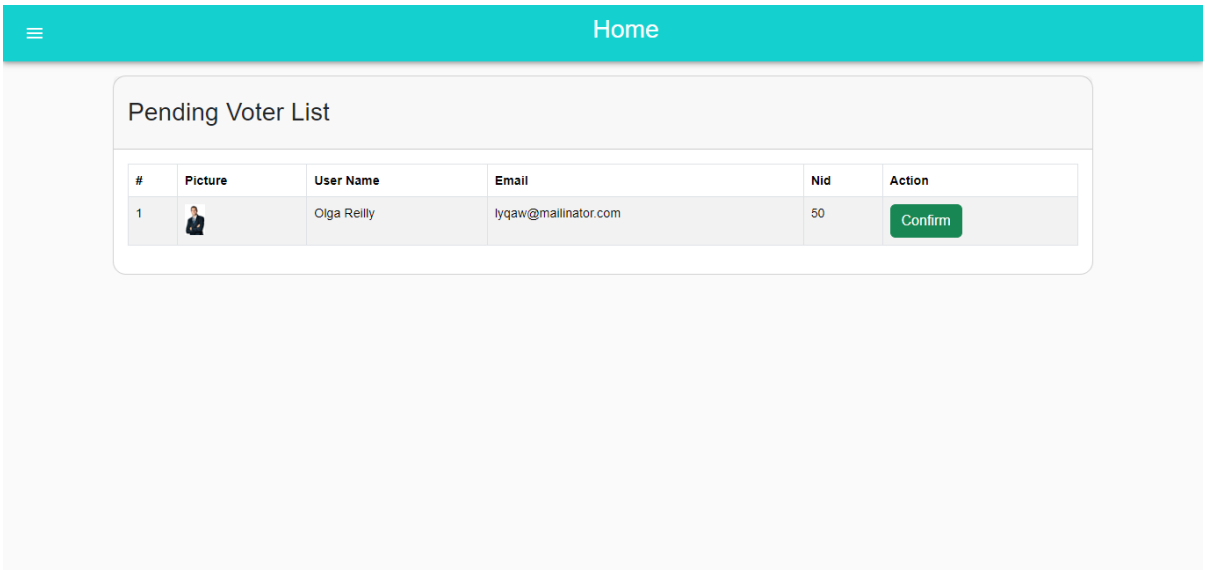


Fig: 5.20 Pending Voter

Election Result

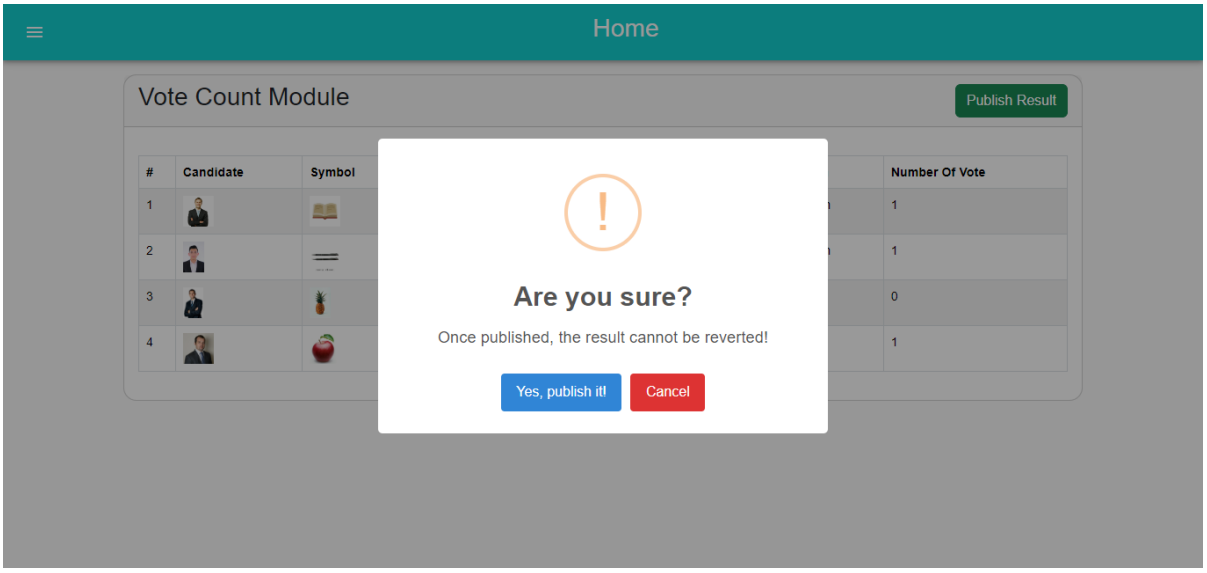


Fig: 5.21 Election Result

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion

Online voting systems have been a topic of discussion and debate for many years, and there are various arguments both for and against their adoption.

One of the main arguments in favor of online voting is the increased accessibility and convenience it can provide. Online voting can make it easier for people to participate in the electoral process, particularly for those with disabilities, people living in remote areas, and those who may not have the time or ability to physically travel to a polling station. Online voting can also save time and reduce costs associated with traditional paper-based voting methods.

However, one of the main concerns with online voting is the potential for security breaches, hacking, and fraud. Online voting systems must be designed with strong security measures to protect against tampering, manipulation, and other forms of interference. They also need to ensure that voters' privacy is protected and their votes are accurately counted.

Another concern is that online voting systems may not be accessible or equitable for all citizens. There may be technological barriers for some people, such as those who lack access to reliable internet or technology, which could create an unfair disadvantage for certain groups of voters.

Additionally, the implementation of an online voting system would require significant investment in infrastructure, technology, and cybersecurity measures. There would also need to be significant public education and outreach efforts to ensure that voters are informed about the process and feel confident in the security of the system.

Overall, online voting systems can offer several benefits, including increased accessibility, convenience, and efficiency. However, appropriate security measures must be in place to protect against the risks of fraud and hacking. Any implementation of an online voting system would need to be carefully planned, tested, and monitored to ensure that it is secure, reliable, and equitable for all citizens.

6.2 Scope for Further Developments

There are several areas where further developments can be made to improve the online voting system. Some of the potential areas for development include

1. Improved security measures: Further research and development could lead to the implementation of more advanced security measures such as blockchain technology or multi-factor authentication to improve the security of online voting systems.
2. Accessibility: Efforts could be made to improve the accessibility of online voting systems for people with disabilities, and to reduce the technological barriers that may prevent some citizens from accessing and using the system.

3. Transparency: Further developments could be made to increase the transparency of the online voting system, such as providing real-time updates on the voting process, and making the source code of the system open to the public.
4. Verification: Developments could be made to improve the verification of voters' identities and to ensure that votes are accurately counted and cannot be tampered with.
5. International standards: International standards could be developed to provide a framework for the design and implementation of online voting systems that prioritize security, accessibility, and transparency.

In conclusion, there is significant scope for further developments in the online voting system to improve the security, accessibility, and transparency of the electoral process. With continued research and development, online voting systems have the potential to increase voter turnout, reduce costs, and improve the overall integrity of the electoral process.

6.3 Conclusion

In conclusion, the advent of online voting systems presents a promising avenue for modernizing electoral processes, offering a range of benefits such as enhanced accessibility, convenience, and efficiency. By leveraging digital technologies, online voting has the potential to streamline the voting process, reduce administrative burdens, and accommodate diverse voter needs, thereby fostering greater inclusivity in democratic participation. Moreover, the transition to online voting can yield substantial cost savings and contribute to environmental sustainability by minimizing paper usage and transportation logistics associated with traditional polling methods.

However, the widespread adoption of online voting also introduces inherent challenges that must be diligently addressed. Foremost among these are concerns regarding the security and integrity of the electoral process. The susceptibility of online systems to cyber threats and manipulation underscores the critical importance of implementing robust security measures, stringent authentication protocols, and comprehensive auditing mechanisms to safeguard against potential breaches and fraudulent activities.

Furthermore, safeguarding voter privacy and ensuring the anonymity of ballots are paramount considerations that necessitate stringent data protection measures and transparent governance frameworks. Additionally, the imperative to maintain public trust in the electoral process demands thorough testing, rigorous oversight, and ongoing evaluation of online voting systems to detect vulnerabilities and mitigate risks effectively.

While online voting holds immense promise for modernizing democracy and increasing civic engagement, it is imperative to approach its implementation with caution and foresight. A balanced approach that prioritizes both innovation and security is essential to realize the full potential of online voting while safeguarding the integrity and legitimacy of democratic elections. Ultimately, sustained collaboration among policymakers, technologists, election officials, and stakeholders is indispensable in navigating the complexities of online voting and advancing democratic principles in the digital age.

REFERENCE

[1]David_MeyersMastering MERN Full Stack Web Development with MongoDB Express React and Node," O'Reilly Media, 1st ed., USA, 2020

[2] Jennifer_Clark-Building_Scalable_Web_Applications_with_the_MERN_Stack," Packt Publishing, 1st ed., UK, 2022

[3] Andrew_Smith-MERN Quick Start Guide Build web applications with MongoDB Express js React_and_Node," Packt Publishing, 1st ed., USA, 2019

[4] Rajesh_Gupta-Practical MERN Stack Development Learn MERN stack development by building_real_world_applications," Apress, 1st ed., India, 2023

[5] Sanjay_Patel-The_Complete_MERN_Stack_Developer_Bootcamp," Packt Publishing, 1st ed., USA, 2021

[6] Rachel Williams-MERN Stack Development Step by Step Guide to Build Full Stack Applications with MongoDB, Express.js, React, and Node.js," Independently published, 1st ed., Canada, 2022

[7] Michael Jones-MERN Stack Projects Build 10 Real Web Apps with the MERN Stack from Scratch," Packt Publishing, 1st ed., UK, 2020

[8] Laura Martin-Learn MERN Stack Development Build a Full Stack Application with MongoDB, Express.js, React, and Node.js," Packt Publishing, 1st ed., USA, 2023

[9] Kevin Robinson-MERN Stack Web Development The Complete Bootcamp," Packt Publishing, 1st ed., UK, 2021

[10] Jessica Cooper-MERN Stack Development Guide Your Ultimate Solution to Building Full Stack Applications with MongoDB, Express.js, React, and Node.js," Independently published, 1st ed., USA, 2020