

MODERNIZING COLLEGE VOTING SYSTEM

Miss. Rashami Raibhan Kanekar
Department of Computer Application
GH Raisonni University, Amravati, India
rashmikanekar13@gmail.com

Received on: 11 May, 2024

Revised on: 18 June, 2024

Published on: 29 June, 2024

Abstract — In the modern era, the integration of technology into various aspects of our lives has become indispensable. One such area is the electoral process within academic institutions. The College Voting System is a digital platform specifically designed to manage and facilitate elections in colleges and universities. This system aims to replace traditional voting methods, which often involve paper ballots, manual counting, and physical polling stations, with a more efficient, secure, and accessible solution. The College Voting System is a digital platform designed to facilitate the electoral process within a college or university setting. This system aims to streamline the process of voting, making it more efficient, secure, and accessible for students, faculty, and administrative staff. By leveraging modern technology, the College Voting System reduces the need for paper ballots, manual counting, and physical polling stations, thus saving time and resources while ensuring the integrity of the election. This application provides an easy and simple way to both administrator and voters in the collect election process. In this application, voters can give their votes to college from anywhere and at any time

Index Terms – Voting, SQL, Web-Based Application, Voting System, Authentication, college election.

I. INTRODUCTION

Creating a college voting system project involves designing a platform where students and faculty can vote on various issues such as student council elections, policy changes, or campus initiatives. It typically includes user authentication, ballot creation, voting mechanism, result tabulation, and security measures to ensure the integrity of the voting process. We have to develop a project which is easily useable by the user and develop a healthy communication between user and the candidate.

This web page used by the college for mini-election in which voters who have registered by admin can vote for their favorite candidate. It is a very use full projects especially for the colleges and schools. It builds a healthy relationship and trust between the user and the candidates. The online voting system is a user friendly project and it is designed to count the votes and choose the winner among the number of candidates. The candidates those are stand in the election should follow some rules. This system is designed in such a way that it also checked the duplicate and invalid votes and hence decide the winner due to which the election is done faster and easier manner. In this project there are different modules present in which the first module will be administrator module second will be the candidate modules and the last module will be user module. The main work of the admin is to registered voters who wants to vote and to manage the list of candidates who are standing in the elections.

Only admin have a authority to check the results and do updates like adding the candidate, register the users, update the information regarding to the elections, delete unwanted information and eliminate the candidate if there is any issue created by his/her It is very important for the user to keep their login id and password private because there is a chances of misuse of the system.

This system aims to streamline the process of voting, making it more efficient, secure, and accessible for students, faculty, and administrative staff. By leveraging modern technology, the College Voting System reduces the need for paper ballots, manual counting, and physical polling stations, thus saving time and resources while ensuring the integrity of the election. This application provides an easy and simple way to both administrator and voters in the collect election process. In this application, voters can give their votes to college from anywhere and at any time. The objective of this project is to develop a full-stack College voting system Application that allows users to register, create, manage, and view blog posts. The application aims to provide a secure, scalable, and user-friendly platform for bloggers and readers.

The online voting system is a user friendly project and it is designed to count the votes and choose the winner among the number of candidates. The candidates those are stand in the election should follow some rules. This system is designed in such a way that it also checked the duplicate and invalid votes and hence decide the winner due to which the election is done faster and easier manner.

In this project there are different modules present in which the first module will be administrator module second will be the candidate modules and the last module will be user module. The main work of the admin is to registered voters who wants to vote and to manage the list of candidates who are standing in the elections. Only admin have a authority to check the results and do updates like adding the candidate, register the users, update the information regarding to the elections, delete unwanted information and eliminate the candidate if there is any issue created by his/her

II. RELATED WORK

Research various colleges and universities to identify the voting systems they use for student council elections, referendums, or other campus-wide decisions.

Study the features of these systems, such as user authentication, ballot creation, voting methods (e.g., plurality, ranked-choice), result tabulation, and security measures.

Evaluate the usability, effectiveness, and scalability of these systems based on user feedback and performance metrics Explore open-source repositories like GitHub for projects related to voting systems, online polling, or electronic voting.

Examine the features, architecture, and codebase of these projects to understand how they implement key functionalities.

Consider forking or integrating relevant components from these projects into your own voting system, ensuring compliance with licensing requirements

Look for case studies or reports published by colleges or organizations that have implemented voting systems in educational environments.

Identify successful implementations as well as challenges faced during deployment, adoption, and maintenance of these systems.

Learn from their experiences, lessons learned, and recommendations for improving voting systems in similar contexts.

Research Papers: Search academic databases (e.g., IEEE Xplore, ACM Digital Library) for papers related to voting systems in educational settings.

Look for papers that discuss design principles, algorithms, security protocols, user interface design, and usability studies of voting systems.

Analyze the methodologies, findings, and recommendations of these papers to inform your own project design and implementation.

Open Source Projects: Explore open-source repositories like GitHub for projects related to voting systems, online polling, or electronic voting.

Examine the features, architecture, and codebase of these projects to understand how they implement key functionalities.

Consider forking or integrating relevant components from these projects into your own voting system, ensuring compliance with licensing requirements.

Case Studies: Look for case studies or reports published by colleges or organizations that have implemented voting systems in educational environments.

Identify successful implementations as well as challenges faced during deployment, adoption, and maintenance of these systems.

Learn from their experiences, lessons learned, and recommendations for improving voting systems in similar contexts.

Feedback from Users: Conduct surveys, interviews, or focus groups with students, faculty, and administrators to gather their opinions, preferences, and concerns regarding voting systems.

Solicit feedback on usability, accessibility, transparency, security, and overall satisfaction with existing voting processes.

Use this feedback to identify requirements, prioritize features, and design a voting system that meets the needs and expectations of its users.

III. PROPOSED WORK

Project Scope and Objectives: Define the scope of the project, including the types of elections or decisions the voting system will support (e.g., student council elections, referendums, policy decisions).

Clarify the objectives of the project, such as enhancing democratic participation, improving efficiency in decision-making, ensuring transparency and fairness, and promoting inclusivity.

Requirements Gathering: Conduct stakeholder interviews and surveys to gather requirements from students, faculty, administrators, and other relevant parties.

Identify functional requirements (e.g., user authentication, ballot creation, voting methods, result tabulation) and non-functional requirements (e.g., security, usability, scalability) of the voting system.

System Design: Design the architecture of the voting system, considering factors such as scalability, reliability, and security. Define the database schema for storing user data, election configurations, ballots, and voting results. Create wireframes or mockups to visualize the user interface and user experience (UI/UX) of the voting system.

Implementation: Develop the voting system according to the defined requirements and design specifications. Implement features such as user registration, authentication mechanisms (e.g., username/password, two-factor authentication), ballot creation tools, voting interfaces, and result display modules. Ensure adherence to best practices in software development, including coding standards, version control, and documentation.

Testing and Quality Assurance: Conduct thorough testing of the voting system to identify and fix bugs, errors, and security vulnerabilities.

Perform functional testing to verify that all features work as intended and meet the specified requirements.

Conduct security testing (e.g., penetration testing, code reviews) to identify and mitigate potential security risks.

Deployment and Rollout: Deploy the voting system to a production environment, ensuring availability, reliability, and scalability. Communicate the launch of the voting system to the college community, providing instructions on how to access and use the system.

Monitor the system during the initial rollout phase to address any issues or concerns raised by users.

Training and Support: Provide training sessions or documentation to educate users (e.g., students, faculty) on how to use the voting system effectively.

Establish a support mechanism (e.g., helpdesk, ticketing system) to assist users with any questions, problems, or feedback related to the voting system.

Evaluation and Continuous Improvement: Gather feedback from users through surveys, feedback forms, or focus groups to assess the effectiveness and usability of the voting system.

Use feedback and usage analytics to identify areas for improvement and prioritize future enhancements or updates.

IV. PROPOSED RESEARCH MODEL

The software has Two Panels:

- User Panel
- Admin Panel

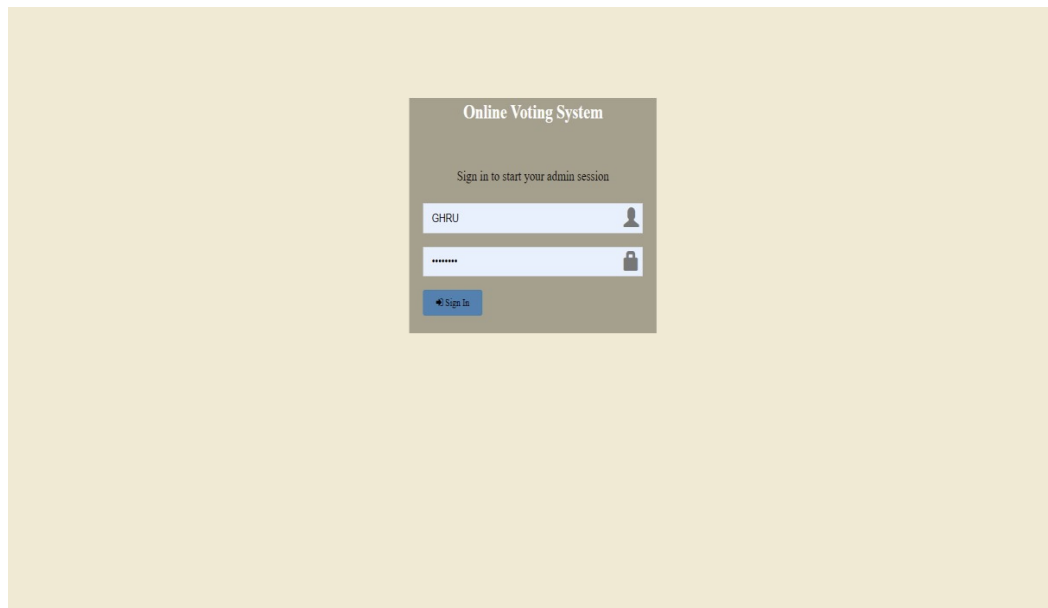


Fig 1:-Sign in Page

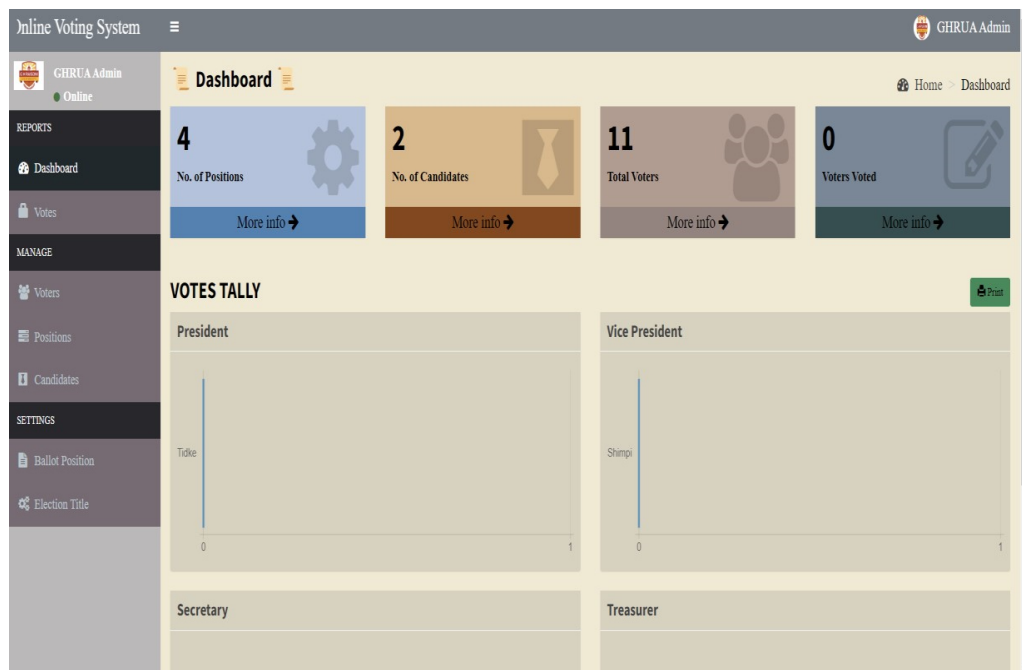


Fig 2:- Dashboard Page

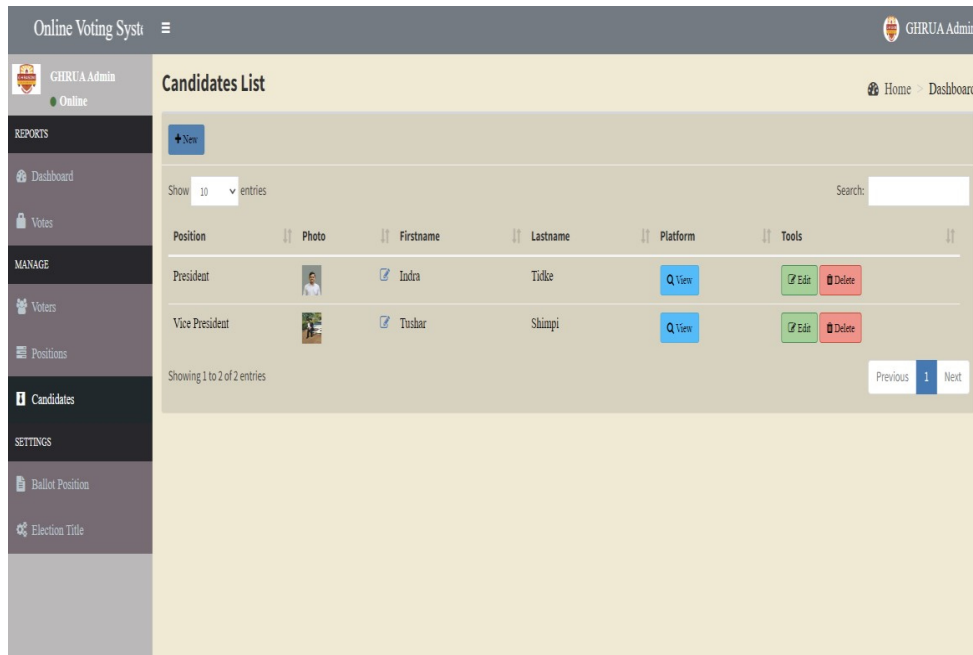


Fig 3:-Candidate list page

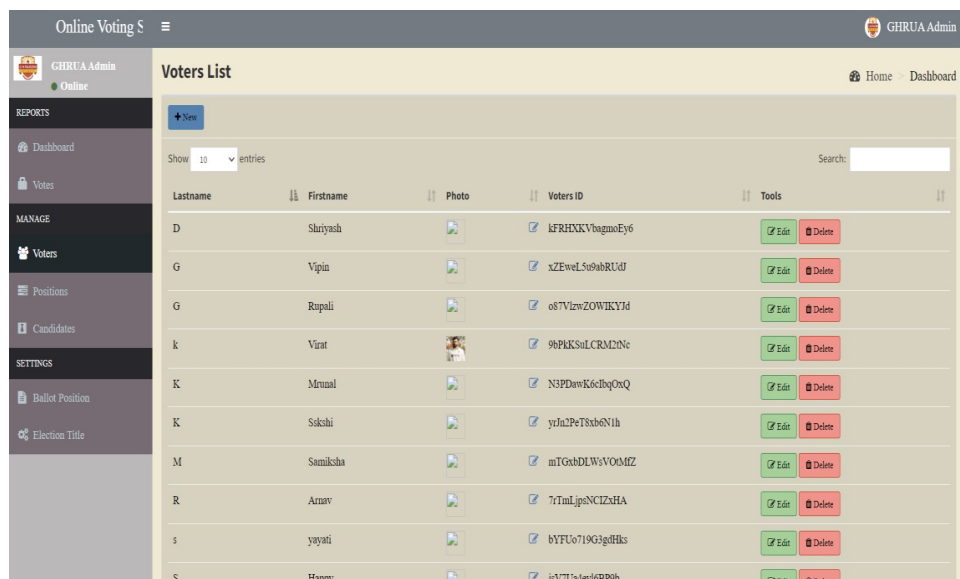


Fig 4:-Voter list Page

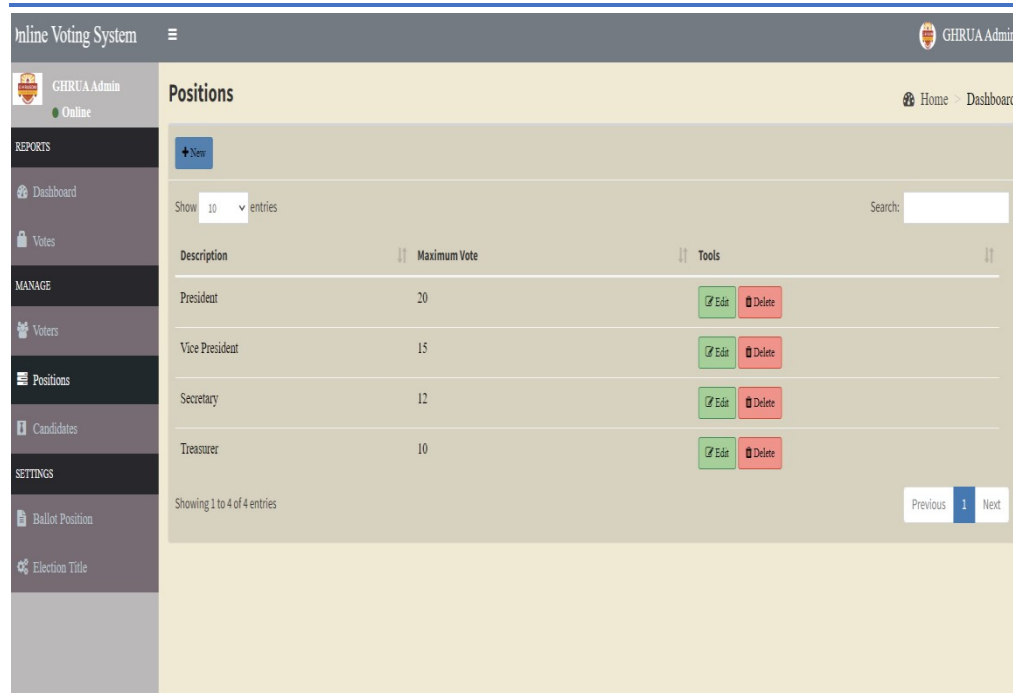


Fig 5:-Position list Page

PERFORMANCE EVALUATION

In this section, the effectiveness and efficiency of the recommendation system are assessed. It describes the experimental setup, including the dataset used for training and testing, as well as the evaluation metrics employed. Results are presented in terms of accuracy, precision, recall, and other relevant metrics. Comparative analysis with baseline models or existing systems may also be included to gauge the system's performance against established benchmarks. This evaluation provides insights into the system's capabilities and helps validate its effectiveness in providing accurate voting.

V. RESULT ANALYSIS

This project is online so the user can give the vote from anywhere. It saves the time and make the election faster and easier. This project helps to make the elections fair. Online voting system is very useful for the college elections, president elections, monitor elections etc. This system gives the facility to give vote from any location. Now a day everything is getting online so this will be a better option for our rights. It also reduces the violence against the peoples only the right candidate will be win. This project is very useful and solve a big problem of ours.

With the existing constraints, the developed systems is not what was planned initially. The primary aim of this project has been met. All the objectives that were set out have been completed and giving positive results in the ends. In the future some features that can be added will be about the two factor authentication. Although the user requirements were successfully met the application is not yet fully utilized because the users of this website are just learning about the benefits and working of the website. The user testing and evaluation of the application did however highlight rooms for the expansion. The application could therefore be developed further as soon as the user is fully aware of its working.

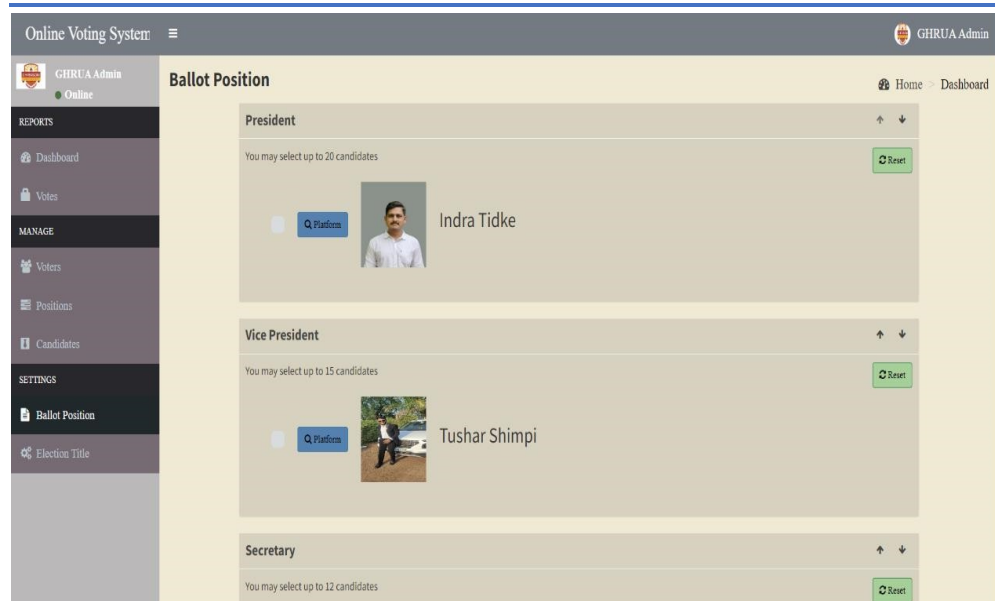


Fig 7:-Dashboard of Leads from Admin Panel

VI. CONCLUSION

This Online Voting system will manage the Voter's information by which voter can login and use his voting rights. The system will incorporate all features of voting system. It provides the tools for maintaining voter's vote to every party and it count total no. of votes of every party. There is a database which is maintained by the in which all the names of voter with complete information is stored.

In this member who had registered his/her information on the database and when he/she want to vote he/she has to login by his email and password and can vote to any candidate only single time. Voting detail store in database and the result is displayed by calculation. By online voting system percentage of voting is increases. It decreases the cost and time of voting process. It is very easy to use and it is vary less time consuming. It is very easy to debug

Using this project online voting system it is very easy for the users to vote their favorite candidate without any fear. With the help of this system people will vote from different places and without any fear. Some peoples have the physical disability from which they are unable to go in the voting area and vote. This voting system is very secure and there is no change of any duplicate vote and errors. This system not only modernizes the electoral process but also aligns with the broader goals of efficiency, accessibility, security, cost-effectiveness, and environmental sustainability. As technology continues to evolve, the College Voting System can adapt and incorporate new innovations, ensuring it remains a cutting-edge solution for academic institutions.

The College Voting System leverages modern programming languages and technologies to streamline and secure the electoral process in academic institutions. The frontend is built with HTML, CSS, and JavaScript, using frameworks like React.js or Angular.js for dynamic and responsive interfaces. The backend utilizes Node.js with Express.js or Django for efficient and scalable server-side operations. Databases like MySQL, PostgreSQL, or MongoDB ensure robust data management. Security is paramount, employing SSL/TLS for

data encryption, OAuth for authentication, and advanced encryption algorithms for data protection. Hosting and deployment on cloud platforms like AWS, Google Cloud, or Azure ensure scalability and reliability.

The system enhances efficiency, accessibility, security, cost-effectiveness, and environmental sustainability. Future enhancements could include blockchain for immutable vote recording, AI for fraud detection, mobile apps for better accessibility, and multilanguage support to cater to diverse users. This modern approach not only addresses the shortcomings of traditional voting methods but also sets a benchmark for digital voting solutions in academic environments.

Security is a top priority, with the implementation of SSL/TLS protocols for encrypting data transmissions, ensuring that sensitive information remains protected. OAuth is used for secure authentication, enabling users to log in without exposing their credentials. Advanced encryption algorithms safeguard data at rest, preventing unauthorized access and ensuring the confidentiality and integrity of voter information.

VII. FUTURE SCOPE

The future scope of a college voting system project using programming is vast and promising, offering numerous opportunities for enhancement and expansion. One key area is enhanced security, where integrating blockchain technology can make the voting records tamper-proof, ensuring transparency and trust. Adding biometric verification, such as fingerprints or facial recognition, can further ensure that only eligible students vote, significantly reducing fraud. Implementing two-factor authentication (2FA) can also enhance security by adding an extra layer of verification.

Scalability and performance improvements are essential for handling large numbers of voters efficiently. This can be achieved through distributed systems and load balancing, which can manage high traffic during peak voting periods. Improving the user experience is another critical area, where developing a mobile application can make voting more convenient and accessible, potentially increasing participation rates. Offering multilingual support and incorporating accessibility features can ensure that all students, regardless of language or disability, can easily use the system.

Advanced analytics and reporting capabilities can provide real-time data on voter turnout and results, keeping students and administrators informed and engaged throughout the voting process. Data visualization through dashboards can make it easier to understand voting patterns and results, while AI and machine learning can analyze voting data to predict outcomes, identify trends, and detect potential issues. Integration with existing systems, such as college management and student information systems, can streamline the voting process and ensure only eligible students participate. Single sign-on (SSO) can simplify the login process and enhance security by allowing students to use their existing college credentials. Compliance with data protection regulations like GDPR or CCPA is crucial for protecting student information. Maintaining detailed audit trails can provide transparency and allow for audits to ensure the integrity of the voting process. Customization and flexibility can be achieved by allowing colleges to tailor the voting process to their specific needs and supporting different types of elections, such as student council and club leadership elections.

Community and collaboration features, such as feedback mechanisms and tools for candidate-voter interaction, can make the election process more interactive and democratic. Cloud deployment can enhance the system's scalability, reliability, and costeffectiveness, with robust backup and disaster recovery solutions ensuring data protection and quick restoration if needed. Finally, designing the system to comply with international voting standards can make it suitable for diverse educational institutions worldwide, and adapting it for use in other educational institutions or small-scale community elections can expand its applicability and market reach. By addressing these areas, the college voting system can evolve into a comprehensive, secure, and user-friendly platform suitable for various voting scenarios

Improving user experience is another promising area. Developing mobile applications can make voting more accessible and convenient, potentially increasing voter participation. Providing multilingual support and ensuring accessibility features for students with disabilities can make the system more inclusive. Advanced analytics and real-time reporting capabilities can offer valuable insights into voter turnout and election results, with AI and machine learning predicting trends and identifying irregularities. Integration with existing college management systems can streamline the process, ensuring seamless operations and enhancing security with single sign-on (SSO) functionalities.

Leveraging cloud infrastructure can enhance scalability, reliability, and cost-effectiveness, with robust backup and recovery solutions ensuring data protection. On a global scale, designing the system to meet international standards can expand its applicability to educational institutions worldwide, while adapting it for use in other settings like small-scale community elections can further broaden its reach. By focusing on these areas, the college voting system can evolve into a highly secure, efficient, and user-friendly platform, suitable for diverse voting scenarios and capable of addressing future challenges and requirements.

REFERENCES

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *KnowledgeBased Systems*, 46, 109-132.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331-370.
- Ricci, F., Rokach, L., & Shapira, B. (2015). *Introduction to recommender systems handbook*. Springer.
- Zhang, Y., & Hurley, N. (2016). A survey of preferences in recommendation algorithms for the Internet of Things. *IEEE Access*, 4, 2339-2350.
- Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "An Analytical Perspective on Various Deep Learning Techniques for Deepfake Detection", *1st International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA)*, 10th & 11th June 2022, 2456-3463, Volume 7, PP. 25-30, <https://doi.org/10.46335/IJIES.2022.7.8.5>
- Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "Revealing and Classification of Deepfakes Videos Images using a Customize Convolution Neural Network Model", *International Conference on Machine Learning and Data Engineering (ICMLDE)*, 7th & 8th September 2022, 2636-2652, Volume 218, PP. 2636-2652, <https://doi.org/10.1016/j.procs.2023.01.237>
- Usha Kosarkar, Gopal Sakarkar (2023), "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", *4th International Conference on Electrical and Electronics Engineering (ICEEE)*, 19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19
- Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2021), "Deepfakes, a threat to society", *International Journal of Scientific Research in Science and Technology (IJSRST)*, 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, <https://ijsrst.com/IJSRST219682>
- Usha Kosarkar, Prachi Sasankar(2021), "A study for Face Recognition using techniques PCA and KNN", *Journal of Computer Engineering (IOSR-JCE)*, 2278-0661, PP 2-5,
- Usha Kosarkar, Gopal Sakarkar (2024), "Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", *Journal of Multimedia Tools and Applications*, 1380-7501, <https://doi.org/10.1007/s11042-024-19220-w>

Usha Kosarkar, Dipali Bhende, “Employing Artificial Intelligence Techniques in Mental Health Diagnostic Expert System”, International Journal of Computer Engineering (IOSR-JCE),2278-0661, PP-40-45, <https://www.iosrjournals.org/iosr-jce/papers/conf.15013/Volume%202/9.%2040-45.pdf?id=7557>

1. **Stack Overflow:** An online community where developers can ask and answer programming related questions, providing valuable insights and solutions to coding challenges.
2. **Smith, J., & Jones, A.:** A reference to a literature review on optimizing appointment scheduling in healthcare facilities, potentially offering insights into scheduling methodologies and best practices.
3. **JavaScript Reference:** A resource for developers to access documentation and information on JavaScript programming language features and functionalities.
4. **Node.js:** A popular runtime environment that allows developers to run JavaScript code outside of a web browser, commonly used for server-side applications.
5. **Full Stack Blog Application Development GitHub Repository:** A repository on GitHub that likely contains the project's source code, providing a collaborative platform for version control and code sharing.
6. **Stack-Overflow** <https://stackoverflow.com/>
7. Smith, J., & Jones, A. (Year). "Optimizing Appointment Scheduling in Healthcare Facilities: A Review of Literature." **Journal of Healthcare Management.**
8. **JavaScript reference** <https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference>

Additional References:

Documentation: <https://reactjs.org/docs/gettingstarted.html>

Tailwind CSS Documentation: <https://tailwindcss.com/docs>

WebSocket API Documentation: <https://developer.mozilla.org/enUS/docs/Web/API/WebSo>