Election System Web App

An SRS document submitted in partial fulfilment of the ECEN 428 course under the supervision of

Dr. Mourad Raafat

Name	ID
Eyad Mohamed	202000041
Sherif Abdelhameed	202000037
Omar Hossam	202000030
Abdelmalek Diaey	202001804

Table of contents

Table of Figures

Introduction

Purpose of the document

The aim of this document is to provide a detailed outline of the process that shall be implemented to run the application by all the stakeholders involved in the development, implementation, and maintenance. This document gives the functional requirements and the non-functional requirements, shows the user interface, database structure and the security measures, testing strategy, deployment procedures and maintenance instructions. Besides, it not only touches on the topics of legal and compliance, but moreover guarantees that compliance with election regulation is ensured. Further, it states the project schedule as the key development phases and the plan of implementation, and it discloses the budget for project launch as well as its maintenance. Maintain continuously. The main purpose of this document is to highlight a detailed plan of action, articulating the vision and understanding all the developers, administrators, and users will be sharing, to ensure the implementation of System web applications.

Scope of the project

The development of the Voting System Web Application project will include writing a code that will use Python as the primary programming language and SQL as the database language. The project which covers building fully an architecture system, its interaction description between components and modules in detail is contained. The application will bring together a set of features that meet most of the needs of the different target groups, which are the voters, who need transparency of the participation process, the administrators, who need a tool to inform voters and the election officials, who need to have control and tools to digitalize their work. We are going to build in features such as secure user identification, candidate registration and electronic voting, to maintain the integrity of the whole election process. Using an SQL connector is a useful way of placing, finding, and processing data into a relational database. Efforts to continue working together with the election officials and legal & compliance experts will remain as the key factor that makes the platform comply with the rules concerning elections and data protection.

Technologies Used

We intend to use HTML, CSS, and Javascript for developing the web app frontend and Nodejs as the website’s backend server connecting it to a SQL database.

Intended Audience

The intended target audience for the election system is eligible voters who have the right to participate in the democratic process by voting. This includes citizens who meet the age, residence, and other legal requirements to participate in elections.

Overview of the document

The document outlines the purpose and scope of a Voting Framework Web Application venture, pointing to be a comprehensive direct for all partners included in its improvement, arrangement, and upkeep. Its subtle elements useful and non-functional necessities, client interface plan, database structure, security measures, testing methodologies, and arrangement forms. Moreover, it addresses legitimate and compliance contemplations to guarantee adherence to race controls. The archive too gives bits of knowledge into extend advance, advancement points of reference, usage plans, and an outline of the budget and assets required for both beginning improvement and ceaseless support. In pith, it serves as a guide cultivating a shared understanding among engineers, chairmen, and clients. The venture scope includes making a user-friendly application utilizing Html, CSS, Javascript, Nodejs and, SQL.

Overall Description

Product Perspective

The product perspective of our election voting system web application highlights this product as a digital platform that is essential for a quality voting process. The web app is built to provide the necessary security and ease of use for the individuals who meet the requirements of voting. The web application is intended to run as a local web application unless we wanted to deploy it.

User characteristics

The user characteristics are one of the essential aspects of the design and implementation of the voting system web application. As the platform recognizes the diversity of demographics and technological proficiency among the voters, inclusivity and accessibility are put at the front line of priorities. Whether they are youthful or old, technologically attuned, or not, our interface is intended to meet the varying needs and interests of all eligible voters, thus contributing to a friendlier and empowering environment.

Constraints

The election voting system web application is subjected to certain limitations that are in the form of regulatory, technological, as well as logistical factors, thus the need to overcome them to ensure the efficiency and compliance of the system. Legal frameworks addressing election issues, as well as scalability requirements are among the main impediments for the platform's design and implementation. Also, the app must operate within the limitations of the user's personal computer, such as hardware and software constraints, to guarantee the best performance and usability. Overcoming these limitations is crucial to making a powerful and first-class election voting app local website that inspires trust and confidence among clients and remains true to the core values of democracy.

Assumptions and dependencies

Assumptions and dependencies of the election voting system local web app are some things that need to be considered to make the process of its operation effective and possible within the boundary of computer user’s environment. The app requires a stable local infrastructure, which depends on the user's PC having sufficient hardware and software resources to ensure efficient functioning. It is also based on the assumption that the PC will be secured being protected from security threats that may lead to the alteration and losing of the voting data.

Functional requirements

Voter Registration Service:

- Allow eligible citizens to register to vote.
- Validate voter eligibility based on citizenship, age, and residency requirements.

- Capture and store voter information securely, including personal details and identification documents.

Candidate Nomination Service:

- Enable qualified individuals to register as candidates for various political offices.
- Validate candidate eligibility criteria such as age, residency, and any legal requirements.

Voting Service:

- Provide a secure and accessible platform for eligible voters to cast their ballots.
- Authenticate voters using unique IDs or secure login credentials.
- Display the list of candidates and offices for the current election.

Vote Tallying and Results Service:

- Aggregate and tally votes cast across all voting precincts or constituencies.
- Ensure the accuracy and integrity of the tallying process through secure algorithms and encryption.
- Generate real-time or periodic updates on election results.

Administrative Tools:

- Provide administrative interfaces for election officials to manage voter registration, candidate nominations, and voting processes.

Accessibility Features:

- Incorporate accessibility features to accommodate voters with disabilities, including visual, auditory, or mobility impairments.
- Provide alternative formats for voting materials such as Braille or audio instructions.

Security Measures:

- Implement robust encryption and data protection mechanisms to safeguard voter information and ballot integrity.
- Conduct regular security assessments and audits to identify and mitigate potential vulnerabilities.

Non-functional dependencies

Usability:

- The system should be easy to use for all voters/users.
- The interface should be intuitive and non-complex.

Reliability:

- The system shall be available for voting during the designated election period without interruptions.
- The system must provide a backup mechanism to recover any failures and ensure continuous operation.

Scalability:

- The system shall be able to handle many concurrent users during peak voting times.
- The system shall be designed to accommodate future increases in the number of users and data volume.

Compatibility:

- The app must be compatible with all commonly used web browsers and devices.
- The system should have Arabic and English versions in its user interface.

Security:

- The system shall comply with relevant security standards and regulations set by the government.
- The system shall have mechanisms in place to detect and prevent fraud or tampering with votes.

System interface (UI)

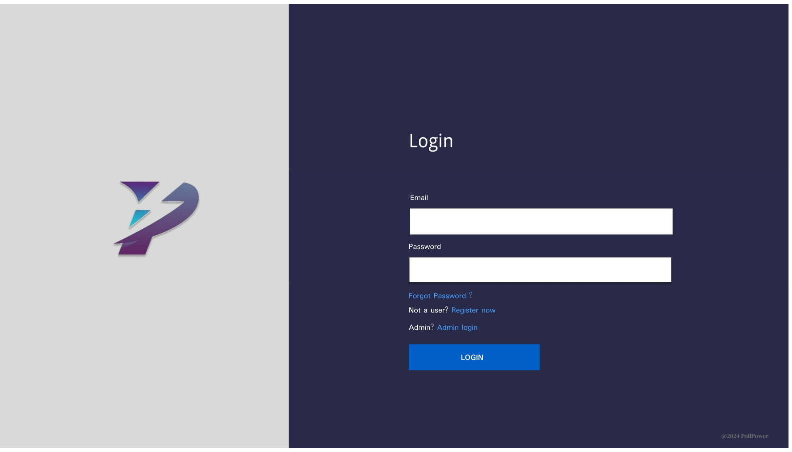


Figure 1 Homepage

This is the web application’s facing homepage, where the user will login to his/her account. There is also an admin login for admins where we will see later on in the document.

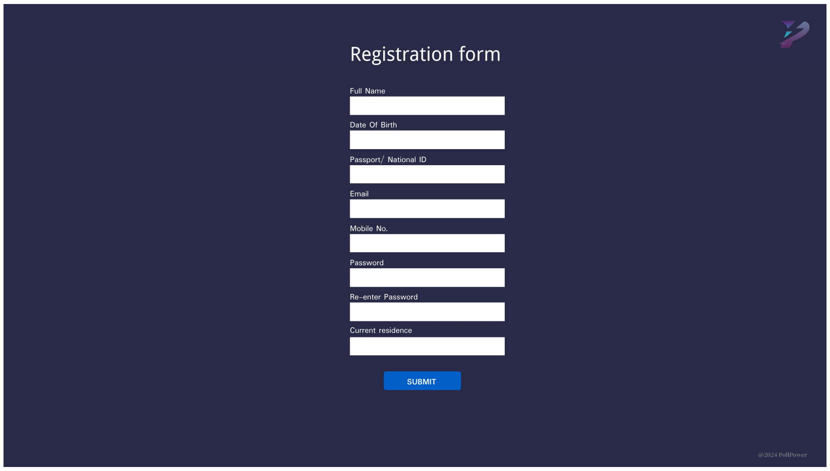


Figure 2 Sign-up form

If it’s the user’s first time using the application, he is expected to sign up for a new account and input all his/her information, in order to proceed.

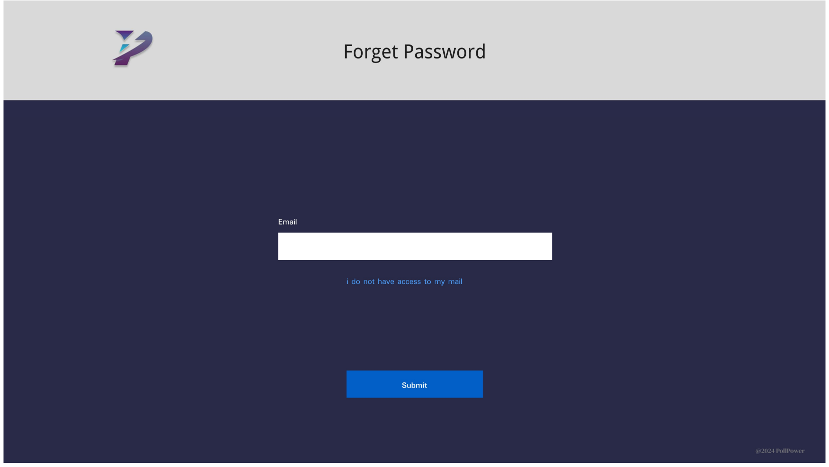


Figure 3 Forget password: Email

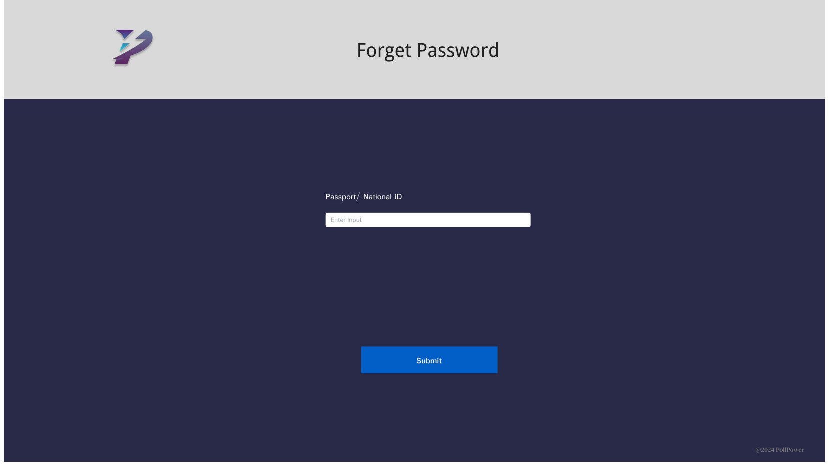
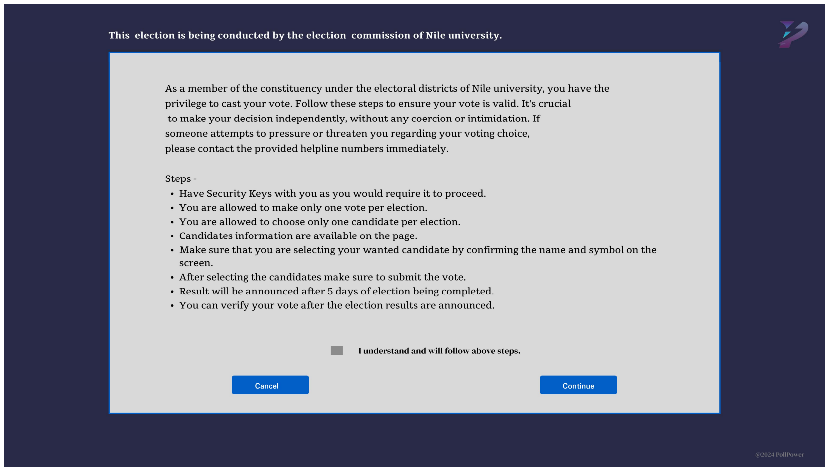


Figure 4 Forget password: National ID

If the user forgot his password, he is required to either input his email or in worst cases input his national ID, if he/she has not had access to their emails.



This election is being conducted by the election commission of Nile university.

As a member of the constituency under the electoral districts of Nile university, you have the privilege to cast your vote. Follow these steps to ensure your vote is valid. It's crucial to make your decision independently, without any coercion or intimidation. If someone attempts to pressure or threaten you regarding your voting choice, please contact the provided helpline numbers immediately.

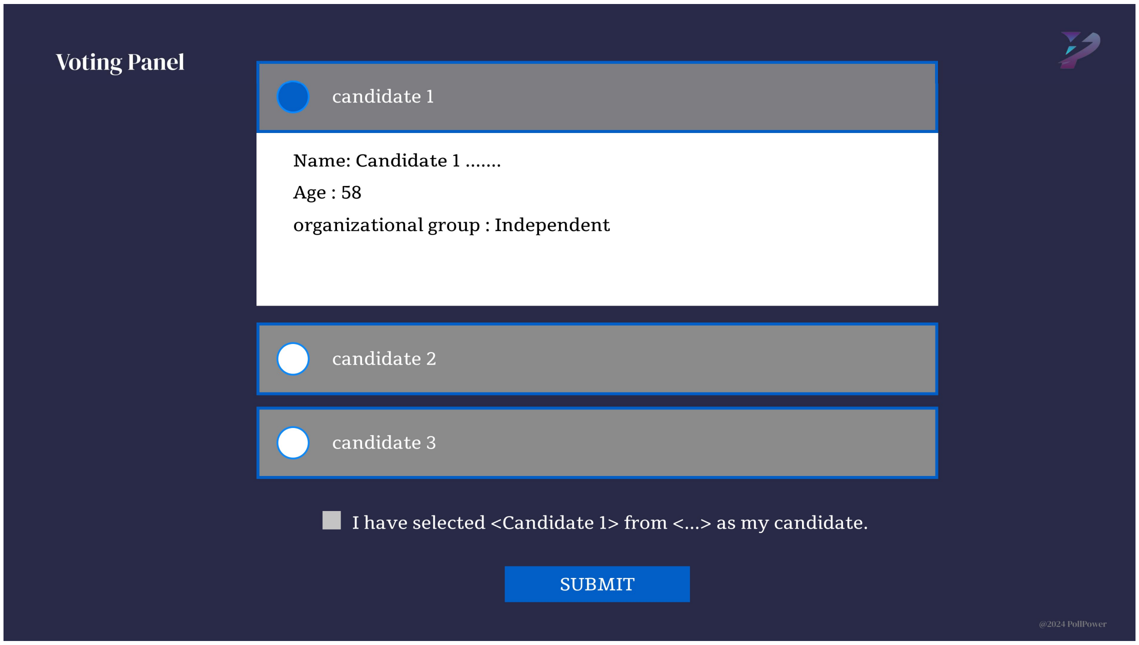
Steps -

- Have Security Keys with you as you would require it to proceed.
- You are allowed to make only one vote per election.
- You are allowed to choose only one candidate per election.
- Candidates information are available on the page.
- Make sure that you are selecting your wanted candidate by confirming the name and symbol on the screen.
- After selecting the candidates make sure to submit the vote.
- Result will be announced after 5 days of election being completed.
- You can verify your vote after the election results are announced.

□ I understand and will follow above steps.

Figure 5 Rules and steps

After proceeding through the login/signup process, the user is notified about the rules and steps he/she must follow, in order to have a smooth voting process.



Voting Panel

● candidate 1

Name: Candidate 1 ....
Age : 58
organizational group : Independent

○ candidate 2

○ candidate 3

□ I have selected <Candidate 1> from <...> as my candidate.

Figure 6 Voting panel

After carefully reading the rules and steps, the user now is in the voting process, where he/she picks the candidate of their choices.

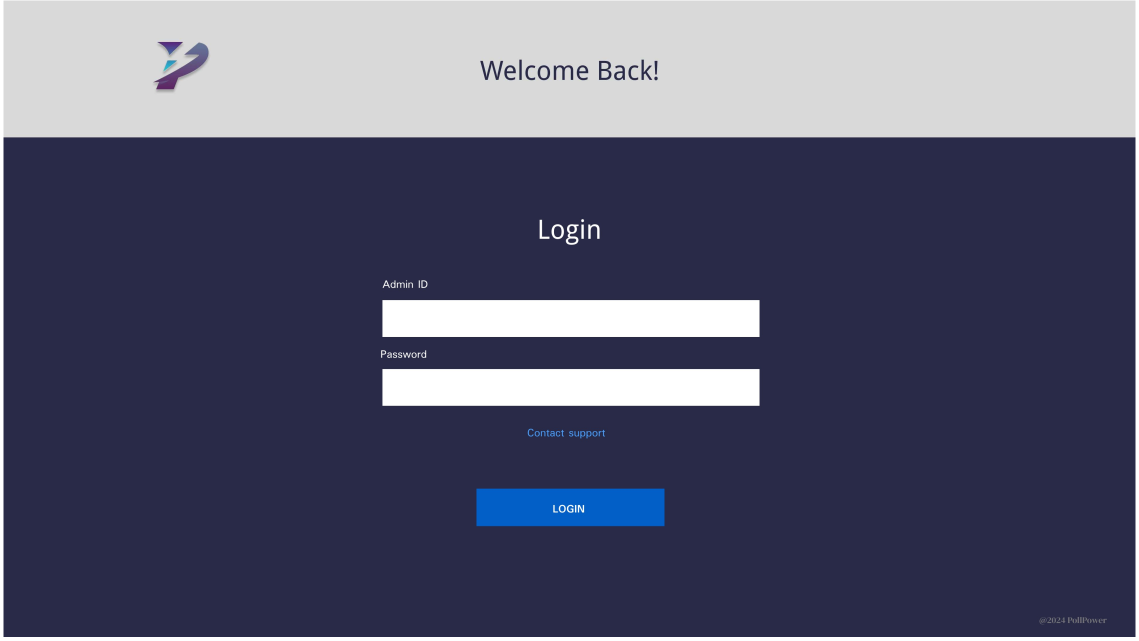


Figure 7 Admins' login

This is the admins’ login page where they are asked to enter their admin ID and password to login.

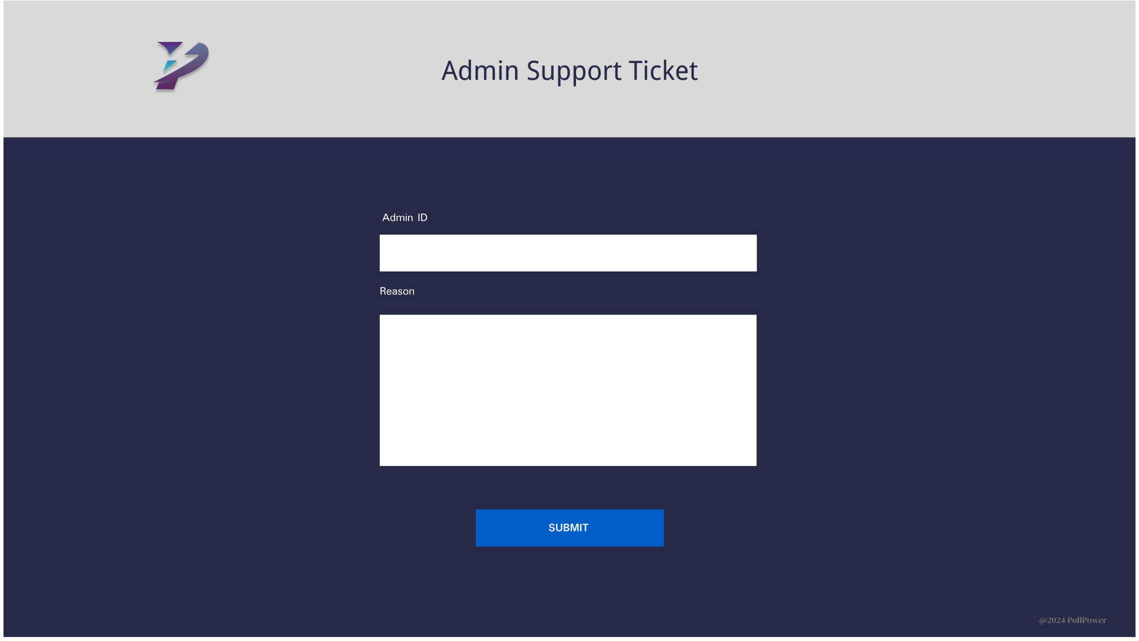
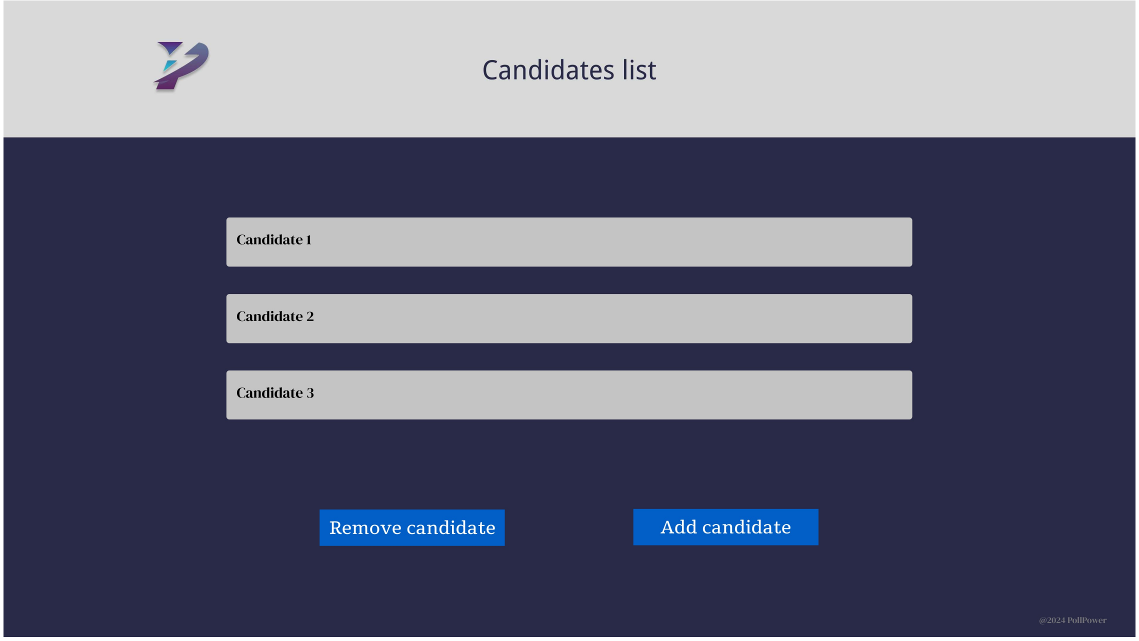


Figure 8 Admins' support ticket form

In times where an admin forgets his password, he has the option to fill out an admin support ticket which will be checked by the developers in order to provide the admins with a new password.



Candidates list

Candidate 1

Candidate 2

Candidate 3

Figure 9 Candidates list

This is the candidates list, where the admins have the full authority and ability to add or remove candidates.

Diagrams

Use case diagram

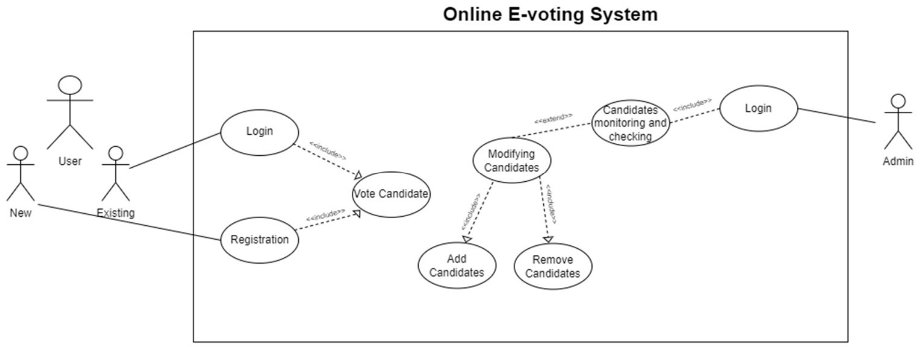


Figure 10 Use case diagram

Use case 1

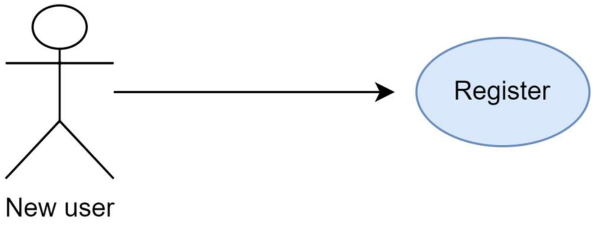


Figure 11 New user registering use case

Title: New user Registration on E-voting system

Actor: New User

Goal: To register for a new account on the E-voting system platform to be able to vote.

Preconditions:

- The user has landed on the homepage of the E-voting system platform.

Main Success Scenario:

1. **Initiation:** The user arrives at the homepage of the e-marketplace platform.
2. **Navigation:** The user locates the "Sign Up" or "Register" button/link prominently displayed on the homepage.

3. **Initiating Registration Process:** The user clicks on the "Sign Up" or "Register" button/link.

4. **Entering Information:**

    - The user is redirected to the registration page where they are prompted to enter their personal information such as full name, email address, password, and any additional required details.

    - The form may include validation checks to ensure the entered information meets the platform's criteria (e.g., valid email format, password length).

5. **Voting agreement process:** The user would be asked to read some rules and accept the terms before heading to the voting panel page.

6. **Voting:** With the voting agreement process complete, the user can now vote to their choice of candidate using the National/Passport id he/she registered with.

Postconditions:

- The user has successfully registered for a new account on the E-voting system platform.

- The user can now access the platform's functionality using their newly created account.

Use Case2

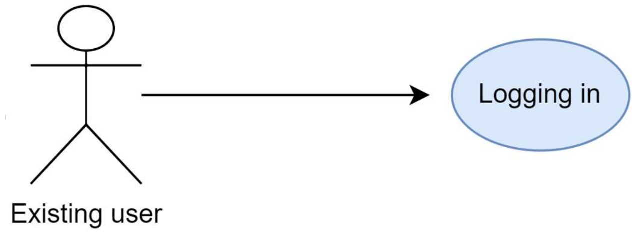


Figure 12 Existing user logging in use case

Title: Existing user Login on E-voting System

Primary Actor: Existing User

Goal: To log in to their account on the E-voting system platform to be able to vote.

Preconditions:

1. The user has previously registered an account on the E-voting system.
2. The user has valid login credentials (email and password).

Main Success Scenario:

1. Initiation:
   - The user arrives at the homepage of the E-voting system platform.
2. Navigation:
   - The user locates the "Log In" or "Sign In" button/link prominently displayed on the homepage.
3. Initiating Login Process:
   - The user clicks on the "Log In" or "Sign In" button/link.
4. Entering Credentials:
   - The user is redirected to the login page where they are prompted to enter their login credentials, typically their email address and password.
   - The form includes validation checks to ensure the entered information meets the platform's criteria (e.g., valid email format, password length).
5. Submitting Information:
   - The user clicks the "Submit" or "Log In" button.
6. Validation:
   - The system checks the entered credentials against the stored data.

- If the credentials are correct, the system logs the user in and creates a user session.

7. Accessing Account:
   - Upon successful login, the user is redirected to the dashboard or home page, where they can access the voting functionalities and other services provided by the platform.
8. Voting agreement process:
   - The user would be asked to read some rules and accept the terms before heading to the voting panel page.
9. Voting:
   - With the voting agreement process complete, the user can now vote to their choice of candidate using the National/Passport id he/she registered with

Extensions:

1. Invalid Credentials:
   - Condition: The user enters an incorrect email or password.
   - System Response: The system displays an error message: "Invalid email or password. Please try again."
2. Forgot Password:
   - Condition: The user clicks the "Forgot Password" link.
   - System Response:
     - i. The system redirects the user to the "Forgot Password" page.
     - ii. The user is prompted to enter their registered email address.
     - iii. The user submits the email address.
     - iv. The system sends a password reset link to the user's email address.
     - v. The user receives the email and follows the instructions to reset their password.

Postconditions:

1. The user is successfully logged into their account on the E-voting system platform.
2. The user can now access the platform's voting functionalities and other services using their account.

Use Case 3

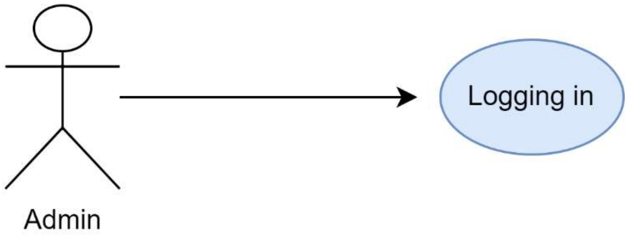


Figure 13 Admin logging in use case

Title: Admin Login on the E-voting System

Primary Actor: Admin User

Goal: To log in to their admin account on the E-voting system platform, manage candidates, and contact support if needed.

Preconditions:

1. The admin has his credentials already stored in the system's database.
2. The admin has valid login credentials (email and password).

Main Success Scenario:

1. Initiation:
   - The admin arrives at the homepage of the E-voting system platform.
2. Navigation:
   - The admin locates the "Admin Log In" or "Admin Sign In" button/link prominently displayed on the homepage.
3. Initiating Login Process:
   - The admin clicks on the "Admin Log In" or "Admin Sign In" button/link.
4. Entering Credentials:
   - The admin is redirected to the admin login page where they are prompted to enter their login credentials, typically their email address and password.
   - The form includes validation checks to ensure the entered information meets the platform's criteria (e.g., valid email format, password length).
5. Submitting Information:
   - The admin clicks the "Submit" or "Log In" button.

6. Validation:
    - The system checks the entered credentials against the stored data.
    - If the credentials are correct, the system logs the admin in and creates an admin session.
7. Accessing Admin Dashboard:
    - Upon successful login, the admin is redirected to the admin dashboard, where they can access functionalities, such as managing candidates and contacting support.
8. Managing Candidates:
    - The admin can navigate to the "Manage Candidates" section.
    - The admin can add a new candidate by entering the candidate's details (name, age) and clicking the "Add Candidate" button.
    - The admin can remove a candidate by selecting the candidate and clicking the "Remove Candidate" button.
9. Contacting Support:
    - The admin can navigate to the "Support" or "Contact Support" section.
    - The admin can fill out a support form with their query or issue and submit it.
    - The system confirms that the support request has been sent.

Extensions:

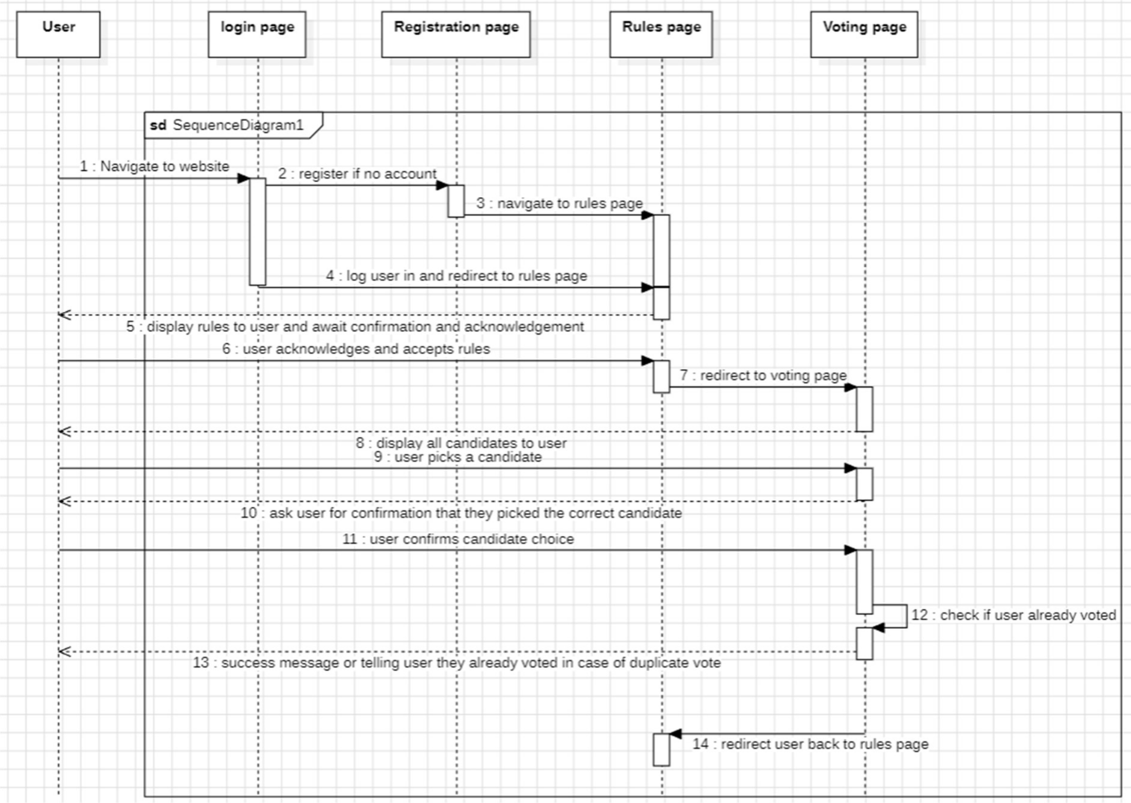
1. Forgot Password:
    - Condition: The admin clicks the "Forgot Password" link.
    - System Response:
        i. The system redirects the admin to the "Forgot Password" page.
        ii. The admin is prompted to enter their registered email address.
        iii. The admin submits the email address.
        iv. The system sends a password reset link to the admin's email address.
2. The admin receives the email and follows the instructions to reset their password.

Postconditions:

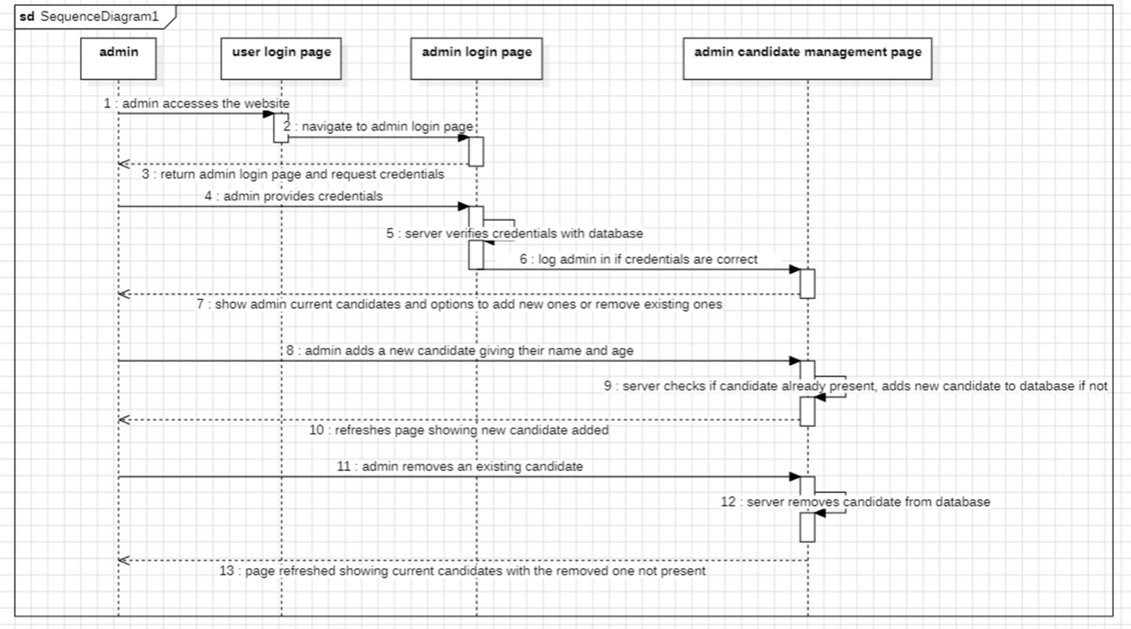
1. The admin is successfully logged into their admin account on the E-voting system platform.
2. The admin can now manage candidates and contact support using their admin account.

Sequence diagrams

Voting a candidate


User
login page
Registration page
Rules page
Voting page
sd SequenceDiagram1
1 : Navigate to website
2 : register if no account
3 : navigate to rules page
4 : log user in and redirect to rules page
5 : display rules to user and await confirmation and acknowledgement
6 : user acknowledges and accepts rules
7 : redirect to voting page
8 : display all candidates to user
9 : user picks a candidate
10 : ask user for confirmation that they picked the correct candidate
11 : user confirms candidate choice
12 : check if user already voted
13 : success message or telling user they already voted in case of duplicate vote
14 : redirect user back to rules page

Admin’s candidates modifying


sd SequenceDiagram1
admin
user login page
admin login page
admin candidate management page
1 : admin accesses the website
2 : navigate to admin login page
3 : return admin login page and request credentials
4 : admin provides credentials
5 : server verifies credentials with database
6 : log admin in if credentials are correct
7 : show admin current candidates and options to add new ones or remove existing ones
8 : admin adds a new candidate giving their name and age
9 : server checks if candidate already present, adds new candidate to database if not
10 : refreshes page showing new candidate added
11 : admin removes an existing candidate
12 : server removes candidate from database
13 : page refreshed showing current candidates with the removed one not present

Test Cases

Test Case ID: TC-001

Test Priority (Low/Medium/High): High

Module Name: registration screen

Test Title: User Registration
Description: Verify that a new user can register successfully.

Test Designed by: Abdelmalek

Test Designed date: 23/5/2024

Test Executed by: Sherif

Test Execution date: 24/5/2024

Pre-conditions: user must not have an already registered phone number, email, and national id.
Dependencies: server.js initialized

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to registration page		Registration page shows up	As expected	Pass	
2	Provide valid email	Email: example@gmail.com	Fields work properly and validate the data	As expected	Pass	
3	Provide valid password	Password: egypt1234	Password should be not visible	As expected	Pass	
4	Fill in the rest of the required information	Random inputs	No errors	As expected	Pass	
5	Click on the register button		User registered and redirected to website	As expected	Pass	

Post-conditions: User has valid username and password

Test Case ID: TC-002	**Test Designed by:** Eyad
Test Priority (Low/Medium/High): High	**Test Designed date:** 23/5/2024
Module Name: login screen	**Test Executed by:** Sherif
Test Title: User Login	**Test Execution date:** 24/5/2024
Description: Verify that a registered user can log in with valid credentials.	

Pre-conditions: User must be registered and have valid credentials.
Dependencies: server.js initialized

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page		Login page shows up	As expected	Pass	
2	Provide valid email	Email: sherifhameed30@gmail.com	Email validation works as intended	As expected	Pass	
3	Provide valid password	Password: egyptegypt	Password validation works as intended	As expected	Pass	
4	Click on Login button		User should be able to login	As expected	Pass	

Post-conditions: User should be successfully logged in and redirected to the voting rules page.

Test Case ID: TC-003	**Test Designed by:** Omar
Test Priority (Low/Medium/High): High	**Test Designed date:** 23/5/2024
Module Name: admin login screen	**Test Executed by:** Eyad
Test Title: Admin login **Description:** Verify that an admin can log in with valid credentials.	**Test Execution date:** 24/5/2024

Pre-conditions: Admin must be registered and have valid credentials.
Dependencies: server.js initialized**,** Ensure the admin credentials are checked against admin database.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to admin login page		Admin login page shows up	As expected	Pass	
2	Provide valid username	user: abdelmalek	Fields work properly and validate the data	As expected	Pass	
3	Provide valid password	Password: shaba7gamed	Password should be not visible	As expected	Pass	
4	Click on Login button		Admin logged in and redirected to admin panel	As expected	Pass	

Post-conditions: Admin session should be active and admin dashboard should be accessible.

Test Case ID: TC-004	**Test Designed by:** Omar
Test Priority (Low/Medium/High): Medium	**Test Designed date:** 23/5/2024
Module Name: user voting screen	**Test Executed by:** Abdelmalek
Test Title: Voting for a Candidate **Description:** Verify that a logged-in user can vote for a candidate.	**Test Execution date:** 25/5/2024

Pre-conditions: User must be logged in and voting session must be active.
Dependencies: server.js initialized

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to the voting page.					
2	Select a candidate from the list of candidates.					
3	Confirm the vote in the confirmation dialog.					
4	Click on the 'Vote' button		Vote should be successfully recorded and a confirmation message should be displayed.	As expected	Pass	Ensure that the user cannot vote more than once.

Post-conditions: User should not be able to vote again, and the vote count for the selected candidate should increase by one.

Test Case ID: TC-005	**Test Designed by:** Abdelmalek
Test Priority (Low/Medium/High): High	**Test Designed date:** 23/5/2024
Module Name: registration screen	**Test Executed by:** Eyad
Test Title: Admin Adding/Removing Candidates **Description:** Verify that an admin can add or remove candidates.	**Test Execution date:** 25/5/2024

Pre-conditions: Admin must be logged in.
Dependencies: server.js initialized

Adding candidates:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to the candidate management page.					
2	Enter the candidate's name.	Joe Biden	Text box works as intended with data validation	As expected	Pass	
3	Enter the candidate's age.	78	Text box works as intended with data validation	Server accepts non-integer values as age.	fail	
4	Click on the 'Add Candidate' button.		Candidate added immediately and shows up in website and database	As expected	Pass	

Post-conditions: Candidate should be successfully added, and the changes should reflect immediately.

removing candidates:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to the candidate management page.					
2	Select a candidate from the list.	Random candidate				
3	Click on the 'Remove' button next to the candidate's name.		Candidate removed immediately and change reflects in website and database	As expected	Pass	

Post-conditions: Candidate should be successfully removed, and the changes should reflect immediately in website and database.

Github Repository link

https://github.com/ohkandil/PollPower.git