Final Report

for

Secure Ranked Choice E-Voting using Shuffle Sum Protocol

Prepared by

Anirudh Achal (191CS108)
Mudit Singhal (191CS138)
Chaithanya Shyam D (191CS218)
Adithya Rajesh (191CS203)
Ayushman Rana (191CS214**)**
Suresh Kamediya (191CS158)
Mihir M Ketkar (191CS229)

NITK Surathkal, Karnataka

1st November 2021

Table of Contents

Introduction 4

Objective 4

Abbreviations 4

References 5

Overview 5

Overall Description 5

Product Perspective 5

Product Functions 5

User Classes and Characteristics 6

Design and Implementation Constraints 6

Assumptions and Dependencies 6

Dataflow diagram 6

DataFlow Diagram Explanation 7

External Interface Requirements 7

User Interfaces 7

Hardware Interfaces 7

Software Interfaces 8

Communications Interfaces 8

System Features 8

User Login and Authentication 8

Election hosting and Admin Privileges 9

User Dashboard 9

Voting in Elections 10

Ranked Choice System 10

Cryptography for secure anonymous votes 11

User Notifications 11

Activity Diagram 12

Activity Diagram Explanation 13

Use Case Diagram 14

Performance Requirements 15

Safety Requirements 15

Security Requirements 15

Software Quality Attributes 15

Implementation tour 16

Testing and maintenance 22

Bugs spotted 22

Unit tests 23

Importance of maintenance 23

Maintenance Attributes 23
- Corrective Maintenance 23
- Preventive Maintenance 24
- Perfective Maintenance 25
- Adaptive Maintenance 25

1.Introduction

1.1.Objective

- Design login and authentication frontpage with the relevant backend for the verification of individual students by their college mail ID. Using the tech stack mentioned in software requirements
- Create a user dashboard in which elections/candidates are visible for the relevant users. (Only a set number of users will be voting for a candidate).
- Implement shuffle-sum protocol which securely encrypts the users votes and is the main driving force for the project. For the underlying cryptography, we will be implementing Damgard-Jurik cryptosystem , which is a threshold cryptosystem that enables the distribution of trust among multiple election authorities
- Implementing a web interface which allows the tabulation of election results and the public posting of those results.

1.2.Abbreviations

Abbreviations Used	Meaning
SRCEV	Secure Ranked Choice E-Voting
TBD	To Be Decided
OTP	One Time Password
CI	Communication Interface
ULA	User Login and Authentication
EAP	Election Admin Privileges
UD	User Dashboard
VE	Voting in Election
RCS	Ranked Choice System
SAV	Secure Anonymous Votes
UN	User Notifications
PR	Performance Requirements

SR	Software Requirements
SeR	Security Requirements
SS	Software Quality Attributes

1.3.References

[1] IEEE-830-1998 - IEEE Recommended Practice for Software Requirements Specification

1.4.Overview

The rest of this document is organized into two sections, the Overall Description chapter and the Specific Requirements chapter. The Overall Description chapter is to give an overview of the functionality of SRCEV, in an informal manner, and is intended primarily for the stakeholders. The Specific Requirements chapter is aimed towards the developers to provide the technical details of the functionality.

2.Overall Description

2.1.Product Perspective

As student elections arrive, There's always a dilemma about who to vote for, not because they are not sure about their preferred candidate, But the backlash they might get from their own group by going against them. Elections which happen with google forms are not secure as our mail ID's need to be entered and the admin can know who voted for who. We eliminate this problem all together by cryptographically encrypting the votes and this maintains utmost privacy. Another issue we are eliminating is the wastage of votes. It happens that we might vote for someone who is nowhere close to winning, hence we do not want our votes to go to waste. With rank choice voting, if our first choice doesn't win, our second choice vote will be considered.

This scales well as elections are not just a college thing, It happens everywhere around the world.

2.2.Product Functions

- *User Login and authentication*
- *Election hosting and Admin privileges*
- *User dashboard*
- *Voting in elections*
- *Ranked choice system*
- *Cryptography for secure anonymous votes*
- *User notifications*

2.3.User Classes and Characteristics

The users of this web application will be the general public of any institution where the election is going to be held. The administrator would be the head of the same institution. The users can request to stand as a candidate for any eligible election and have the ability to vote in any election in which they're eligible.

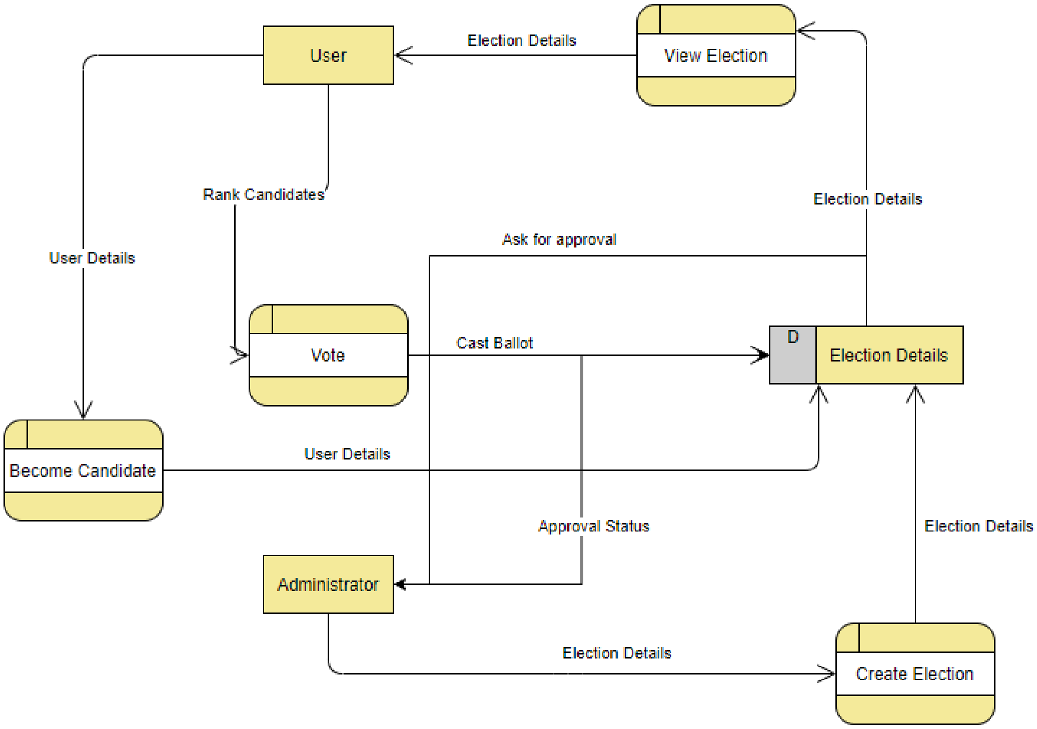
2.4.Design and Implementation Constraints

Web application needs to be compact in size as to provide quick access for the user and hence optimized code which runs on 512 kbps internet. This project needs to be done in a timeline of 2 months to make sure it is present for next semester and we get ample time for testing. No individual module should be straining any other module.

2.5.Assumptions and Dependencies

- *As we are writing the codes on shell, Our database requirement might change on the basis of convenience and performance.*
- *For tallying encrypted votes, we might use table sum protocol instead of shuffle sum protocol on the basis of performance.*

2.6.Dataflow diagram



DataFlow Diagram Explanation

Entities : User and Administrator
Databases : Election Details
Process : View Election, Vote, Create Election, Request to become Candidate

Users are entities who are invited to participate in elections as voters and can also request to become a candidate. They can also view the details of the elections in which they are participating.

Administrators are privileged users who can create elections and also invite participants to the election. They are also responsible for approving user's requests to become candidates in an election.

Election Details is the database that contains all the information about the election. It stores the list of users participating as voters, the list of candidates, time duration of the election, encrypted ballots and the results.

View Election is the action which allows the users to view the details of the election.

Vote is the action through which the voters can rank candidates and cast their ballots.

Create Election is the action through which the administrators can make a new election.

Request to become Candidate is the action through which a voter can ask for approval from the administrators to contest in the election as a candidate.

3.External Interface Requirements

3.1.User Interfaces

UI-1: *For laptop/PC users, the web application shall permit navigation and item selection using the keyboard alone, in addition to using mouse and keyboard combinations.*

UI-2: *For smartphone users, the web application shall support hardware as well as virtual keyboard for textual input, and touch input for item selection.*

3.2.Hardware Interfaces

In decreasing order of priority, the devices to be supported are: smartphones, laptops, personal computers, and tablets. This priority order has been decided keeping in mind the individuals living in remote areas, who have access to low-end devices like smartphones as opposed to high-end devices.

3.3.Software Interfaces

A fully-working internet browser is needed to run the website, preferably Google Chrome/ Microsoft Edge.

3.4.Communications Interfaces

CI-1: The system shall send an email notification to the user when an election is created, and when the election window starts.
CI-2: The system shall send an email message to the user to confirm registration with the system.

4.System Features

4.1.User Login and Authentication

4.1.1 Description

Users are required to register on the website with their NITK email address.
During
registration, the users' NITK email address is verified using a one-time password (OTP) sent to their email. Users will have to login to use the website. Additionally, election admins will have separate accounts which will be registered by the website maintainers.

4.1.2 Stimulus/Response Sequences

Stimulus: User request to create an account.
Response: System displays a form to allow user to enter his name, email, password
which shall be used as inputs to create the account and will send a mail to the user's email id to verify his email.

Stimulus: User requests to login if he already has an account.
Response: System will check the user's credentials and let him access the application.

4.1.3 Functional Requirements

ULA-1: The system shall allow users to create an account and verify themselves via OTP.
ULA-2: The system shall allow users to login to their existing account via Email id and Password.

4.2.Election hosting and Admin Privileges

4.2.1.Description

Election admins are appointed by the website maintainers and superusers. When an admin logs into the website, they can create and host a new election. They can also invite different users registered on the website to participate in the new election. Participants of an election can request the election admin make them a candidate in the ongoing election. The admin can approve/disapprove of these requests and this will be displayed publicly.

4.2.2.Stimulus/Response Sequences

Stimulus : User requests to make him candidate for the particular election.
Response: System will approve or disapprove the candidate's request and display it publicly on the election page.

4.2.3.Functional Requirements

EAP-1: The system shall allow maintainers to create admins who can then host elections.
EAP-2: The system shall allow admins to accept users to stand as candidates for a particular election.
EAP-3: The system shall display status of candidate applications for ongoing elections.

4.3.User Dashboard

4.3.1.Description

All elections in which a user is participating as a voter/candidate will be visible in the user's dashboard. Users can click on a specific election to visit the election page which will have a separate page displaying all the candidates, voters and the results.

4.3.2.Stimulus/Response Sequences

Stimulus: User requests to visit one of the election pages.
Response: System provides him the election page with pre-populated statistics and results of the election.

4.3.3.Functional Requirements

UD-1: The system shall display all the past elections in a dashboard.
UD-2: The system shall allow users to visit a particular election page, to know the statistics and results, if any of the respective elections.

4.4.Voting in Elections

4.4.1.Description

Once a user has accepted to participate in an election, they can enter the election page/portal where they can see the list of candidates and voters. Once the user has viewed the profiles of the candidates and has made a decision, they can cast their vote. Once the election is complete, the results will be displayed on the web portal.

4.4.2.Stimulus/Response Sequences

Stimulus: User requests to visit the particular election's portal to vote.
Response: System will allow the user to visit the particular election portal if he/she can participate in that particular election.

Stimulus: User requests to cast vote to many candidates according to his preference.
Response: System will allow the user to cast the election according to his/her choice.

4.4.3.Functional Requirements

VE-1: The system shall display all the candidates for the particular election that can be voted.
VE-2: The system shall allow users to cast their vote in a particular election, on ranked-choice method

4.5.Ranked Choice System

4.5.1.Description

Voters can rank all the candidates based on their order of preference. If a candidate wins a majority of first-preference votes, he or she is declared the winner. If no candidate wins a majority of first-preference votes, the candidate with the fewest first-preference votes is eliminated. First-preference votes cast for the failed candidate are eliminated, lifting the second-preference choices indicated on those ballots. A new tally is conducted to determine whether any candidate has won a majority of the adjusted votes. The process is repeated until a candidate wins an outright majority.

4.5.2.Functional Requirements

RCS-1: The system shall calculate the result on the basis of the ranked choice algorithm and declare the winner.

4.6.Cryptography for secure anonymous votes

4.6.1.Description

It is crucial that the election is implemented by a system that is secure, confidential, fair, and free of coercion so we are going to implement the Shuffle-Sum protocol which securely tallies encrypted votes in a single transferable vote (STV) ranked-choice election. For the underlying cryptography for Shuffle-Sum, we will implement the Damgard-Jurik cryptosystem, which is a threshold cryptosystem that enables the distribution of trust among multiple election authorities.

4.6.2.Functional Requirements

SAV-1: The system shall encrypt all the incoming votes which basically hides user details of the ballot, to maintain anonymity.

4.7.User Notifications

4.7.1.Description

Users will be notified through email about invitations to elections, status of their vote and final election results.

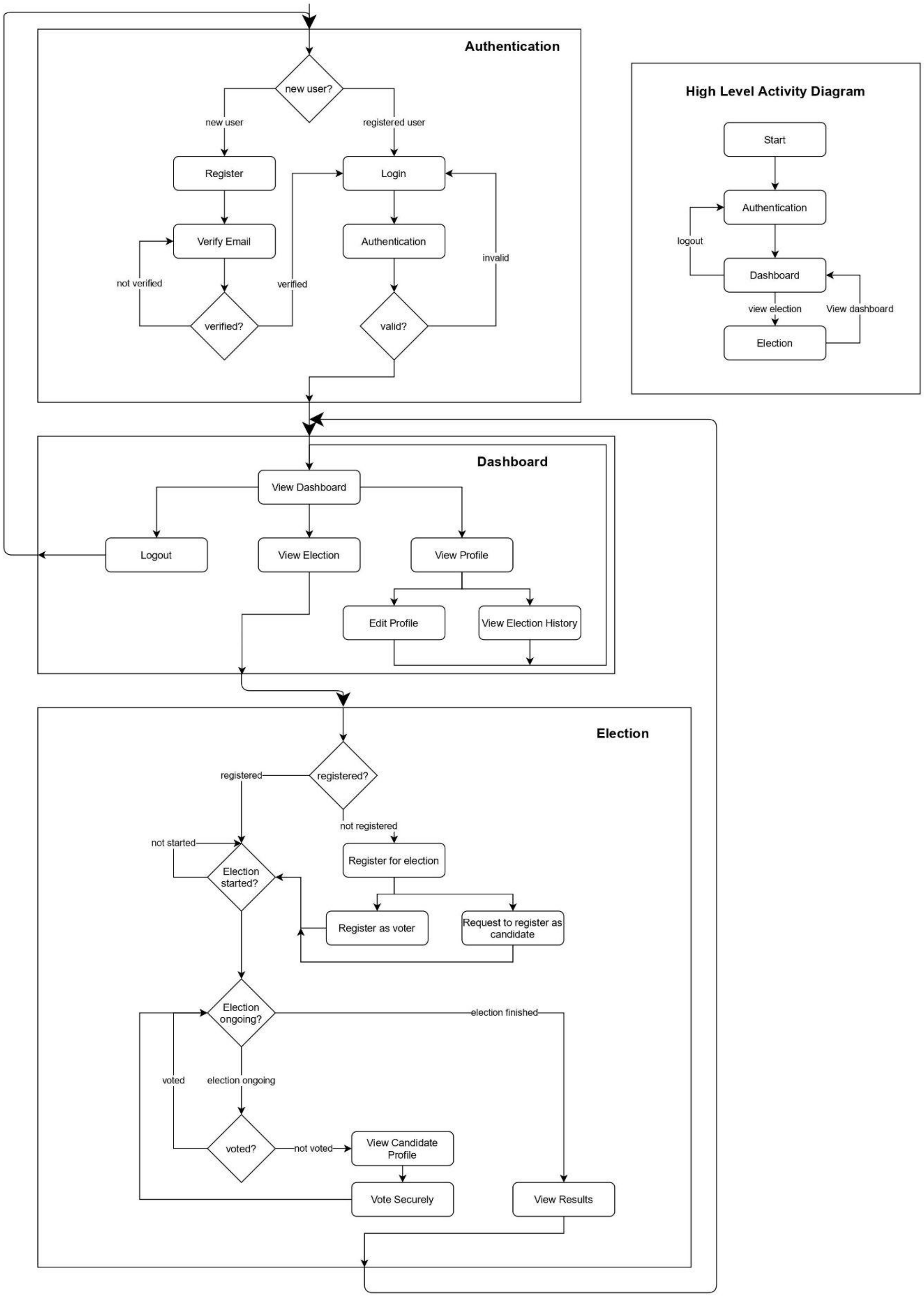
4.7.2.Stimulus/Response Sequences

Stimulus: Users accept the invitation of a particular election.
Response: System will mail him updates(his status and final election results) regarding the particular election.

4.7.3.Functional Requirements

UN-1: The system shall send email notifications if and when any election is created
UN-2: The system shall send his voting status and the final election results when the election ends.

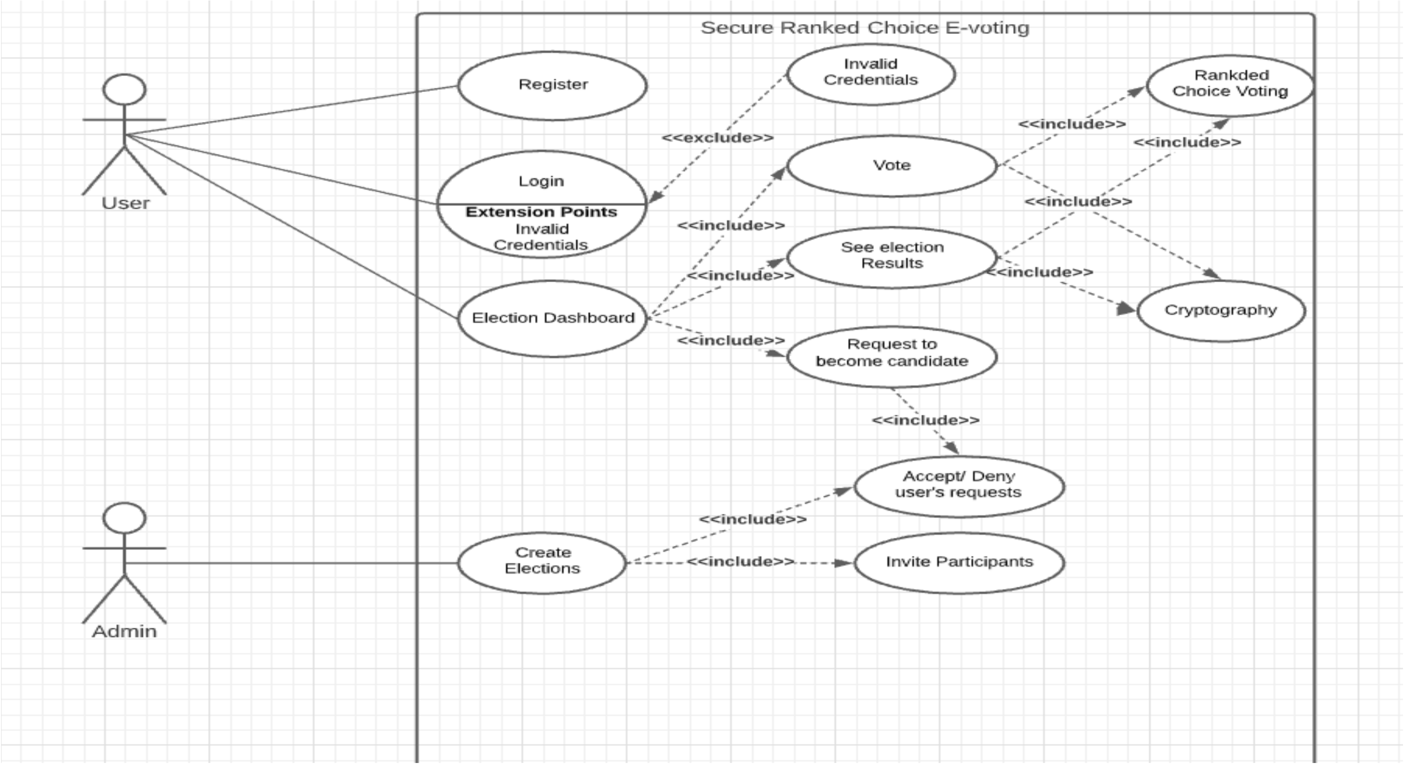
4.8.Activity Diagram



Activity Diagram Explanation

- **High-level:** The high-level activity diagram shows the main activity flow of the application. When users first open the application, they have to first get authenticated. Once the users are authenticated, they can visit the dashboard. From the dashboard, users can view a specific election that they are a part of. This will take the users to an election page where they can view candidates and vote. Users can also log out from the dashboard. This will take them back to the authentication page.
- **Authentication:** When the user enters the authentication module, they can log in to the website if they are a registered user. If the user is not registered, they will need to register. The user's email will need to be verified following the registration. Once this is done, the registered user can log in to the website.
- **Dashboard:** Once a user is authenticated they will be directed to their dashboard. Users can view their profile, view an election or log out. If they choose to view their profile, they will be redirected to a profile page that can be edited and also they can view their election history. If they choose to view an election, they will be redirected to the election page. Otherwise, if the user decides to log out, they will be redirected back to the authentication module.
- **Election:** When a user views an election page, if they are already registered, they can view the different candidate profiles until the election starts. Once the election starts, they can cast their vote securely using rank choice. If the user is not already registered in the election, they will have the option to register as a voter or to request to register as a candidate. Once they have been registered they can vote in the ongoing election. Once the election is complete, the election page will store the results of the election that can be viewed by all voters and candidates.

4.9.Use Case Diagram



Explanation

- It shows the high-level implementation of our application.
- It describes the functional requirements of the project (the way in which actors(like user and admin) interact with the system).
- It helps us to see the different roles of user and administrator.

Actors - User And Admin (Administrator)

Admin role

- He can create new elections and can invite participants(voters) to take part in the election.
- He can also accept/deny user's requests to become a candidate.

User role

- Users can create an account or login to the application.
- Users can also request to become a candidate for a particular election.
- Users can vote(in ranked choice manner), and can see election results.

5.Other Nonfunctional Requirements

5.1.Performance Requirements

5.1.1.PR-1: Web pages must be fully downloadable in no more than 1 second on a 512 kbps connection.

5.1.2.PR-2: On Each webpage, queries made to the database shall take no more than 1 second on a 512 kbps connection.

5.2.Safety Requirements

5.2.1.SR-1: Database must not fail as election data is sensitive and error in it might cause catastrophic results.

5.2.2.SR-2: Algorithms must work perfectly as the whole idea is dependent on it

5.3.Security Requirements

5.3.1.SeR-1: User(voter) details are cryptographically encrypted which enables users to vote freely and admins cannot see who voted for who.

5.3.2.SeR-2: Users cannot see who other users voted for.

5.3.3.SeR-3: Without admin privileges, user cannot create their own elections

5.4.Software Quality Attributes

5.4.1.SS-1: The system shall be easy to use for input preparation, operation, and interpretation.

5.4.2.SS-2: The system shall be straightforward and easy for the user to vote

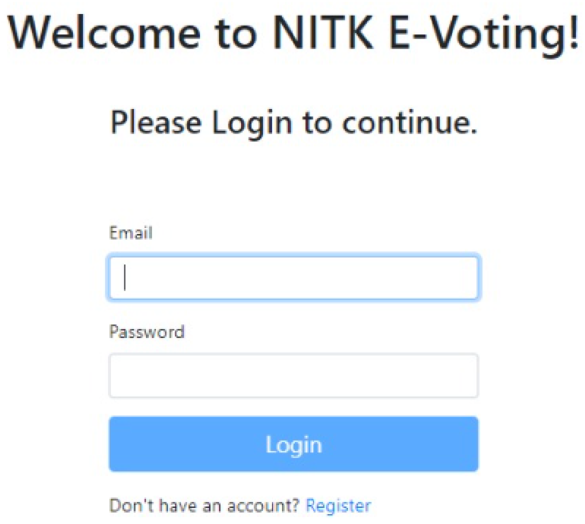
5.4.3.SS-3: The system shall be divided into multiple modules for testing

5.4.4.SS-4: the system shall match the functional requirements.

6.Implementation tour

USER:

When a user first enters the website, they'll be prompted to log in/register. If a user enters the correct email address and password they will be able to access the dashboard.



Welcome to NITK E-Voting!

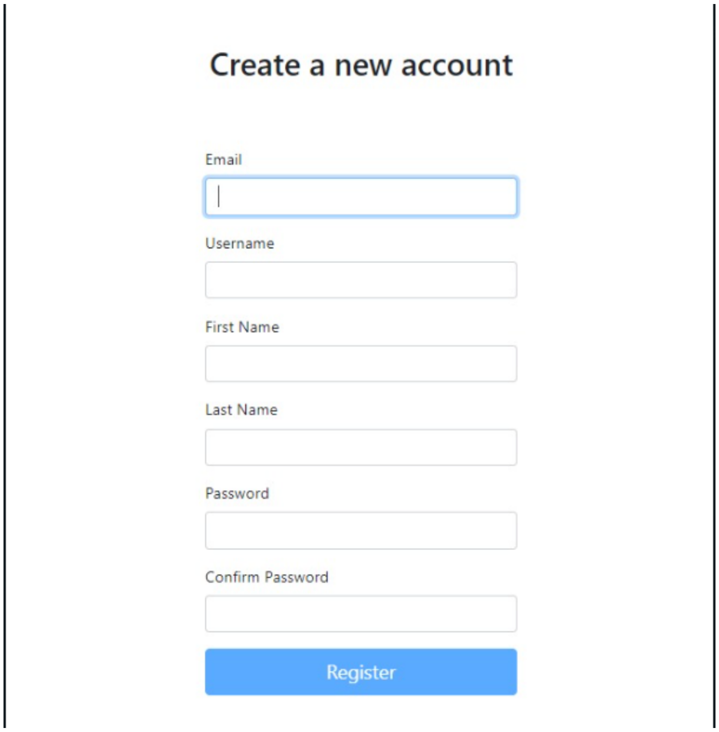
Please Login to continue.

Don't have an account? Register

If you press “Register”, you will be required to enter the information asked below and make sure the email id is working as you will need to verify it.

To create said new account one must enter their name, email address, enter a valid username and assign a password.

An email will be sent to the user email address for verification, failing to do this will result in the account being locked.



Create a new account

Email

Username

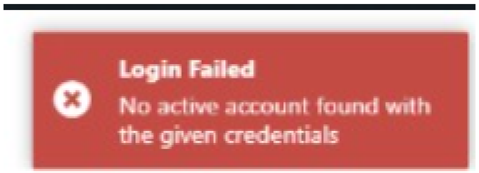
First Name

Last Name

Password

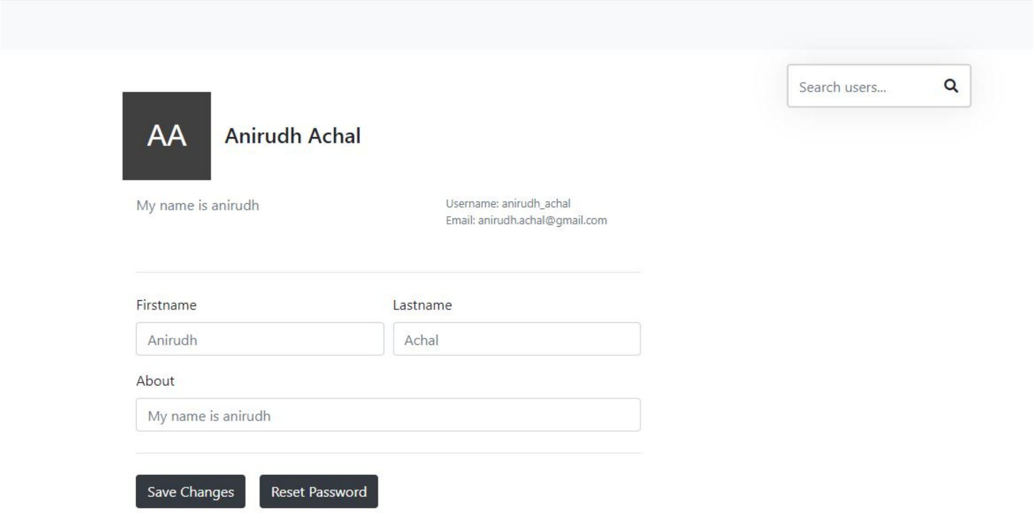
Confirm Password

Wrong password will give this prompt at top right corner



Login Failed
No active account found with the given credentials

There is a user profile page present which gives the user a chance to view their information and change any particular details



Anirudh Achal

My name is anirudh

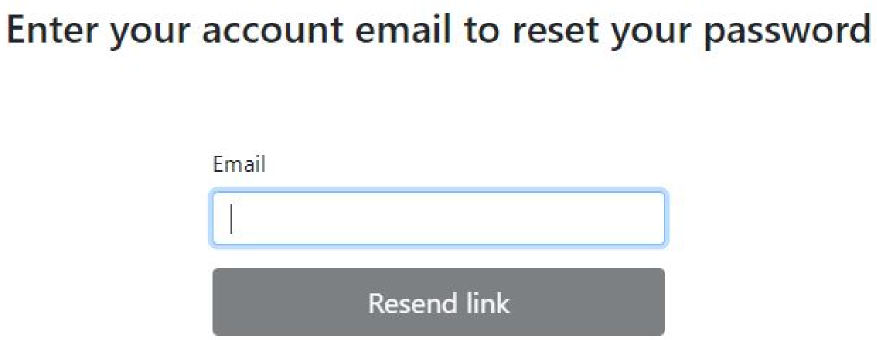
Username: anirudh_achal
Email: anirudh.achal@gmail.com

Firstname: Anirudh

Lastname: Achal

About: My name is anirudh

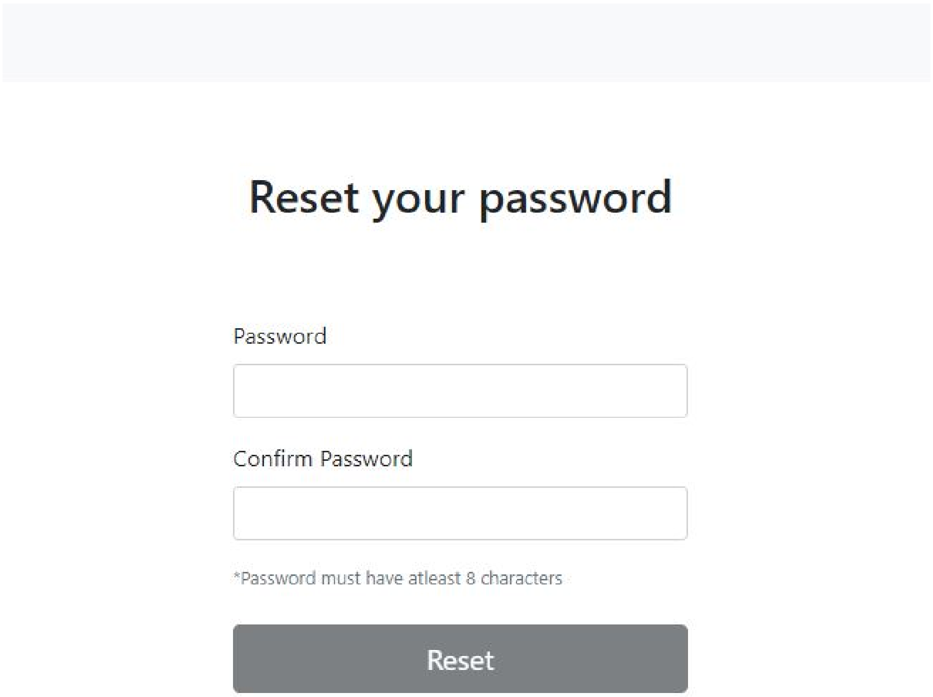
In case you want to change your password, there is an option to change



Enter your account email to reset your password

Email

Resend link



Reset your password

Password

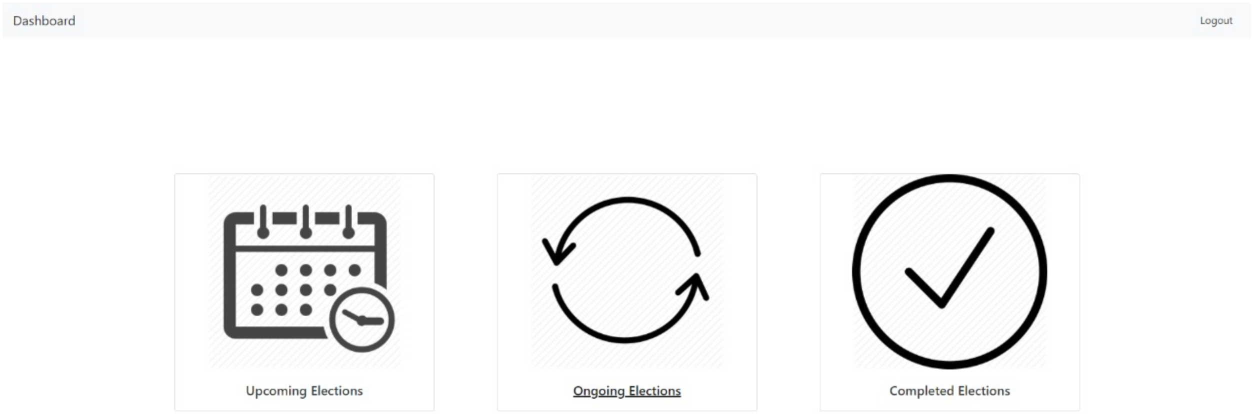
Confirm Password

*Password must have atleast 8 characters

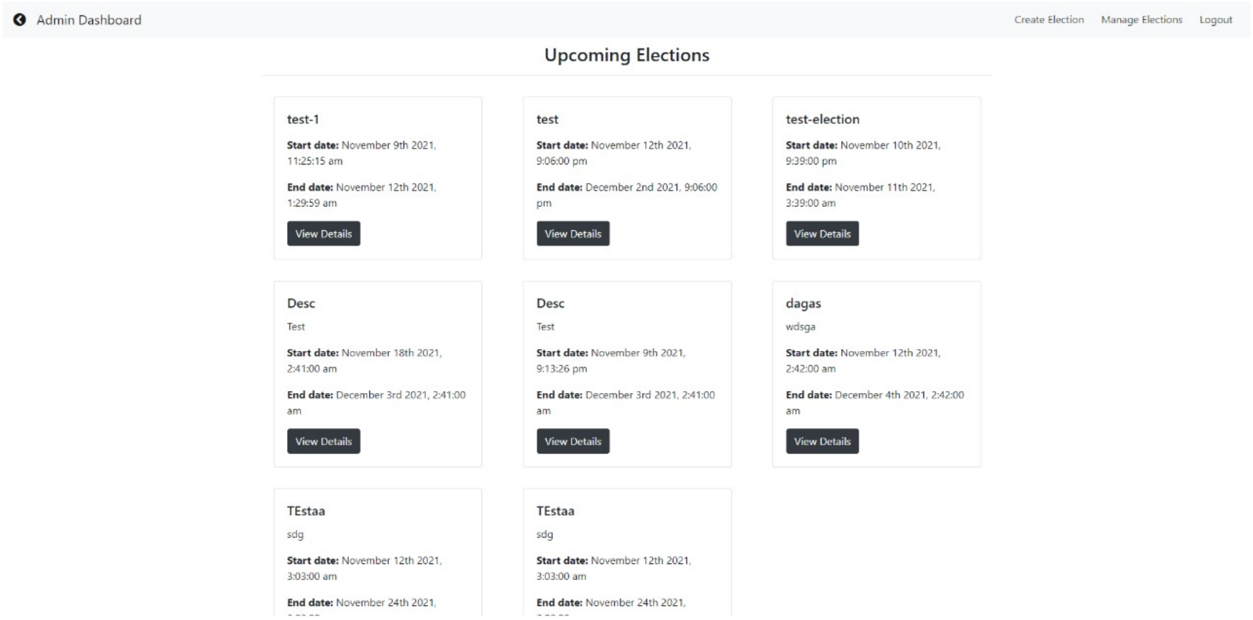
Reset

After logging in, the dashboard will be presented for the user with relevant details to them.

These are the Upcoming elections tab, ongoing and completed elections. As the name suggests, they will redirect you to the appropriate webpage featuring the information about the same.



If you click on “Upcoming Elections”, you’ll see all upcoming elections with the required information. Also an option to stand as a candidate



Upcoming Elections

test-1

Start date: November 9th 2021, 11:25:15 am

End date: November 12th 2021, 1:29:59 am

test

Start date: November 12th 2021, 9:06:00 pm

End date: December 2nd 2021, 9:06:00 pm

test-election

Start date: November 10th 2021, 9:39:00 pm

End date: November 11th 2021, 3:39:00 am

Desc

Test

Start date: November 18th 2021, 2:41:00 am

End date: December 3rd 2021, 2:41:00 am

Desc

Test

Start date: November 9th 2021, 9:13:26 pm

End date: December 3rd 2021, 2:41:00 am

dagas

wdsga

Start date: November 12th 2021, 2:42:00 am

End date: December 4th 2021, 2:42:00 am

TEstaa

sdg

Start date: November 12th 2021, 3:03:00 am

End date: November 24th 2021,

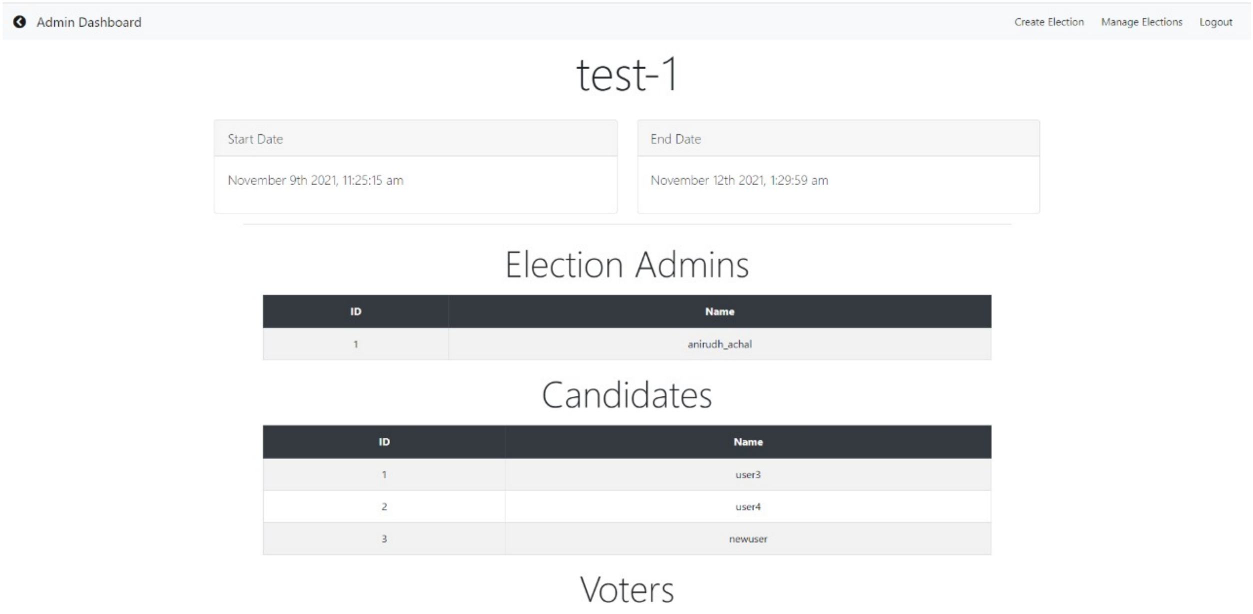
TEstaa

sdg

Start date: November 12th 2021, 3:03:00 am

End date: November 24th 2021,

On the dashboard, clicking on ongoing elections. You will have a chance to vote. You will be eligible to vote in election if and only if you are authorized to by an Admin. This has been done to prevent double-voting.



test-1

Start Date: November 9th 2021, 11:25:15 am

End Date: November 12th 2021, 1:29:59 am

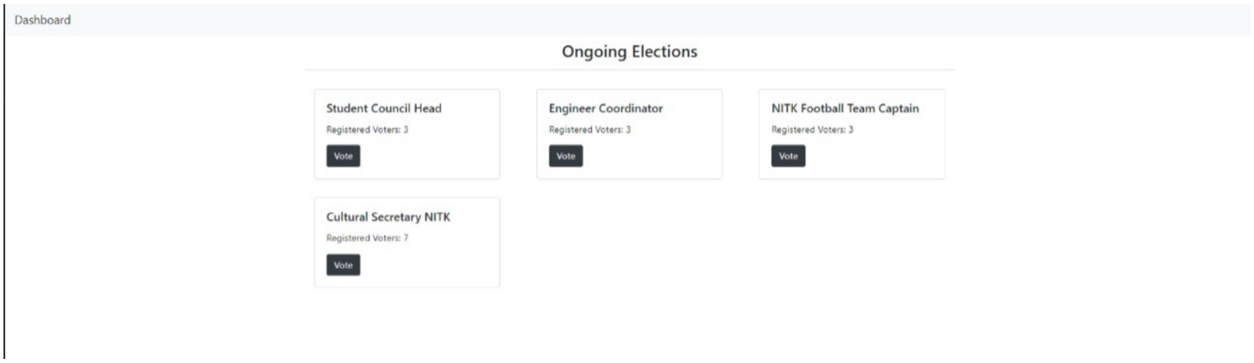
Election Admins

ID	Name
1	anirudh_achal

Candidates

ID	Name
1	user3
2	user4
3	newuser

Voters



Ongoing Elections

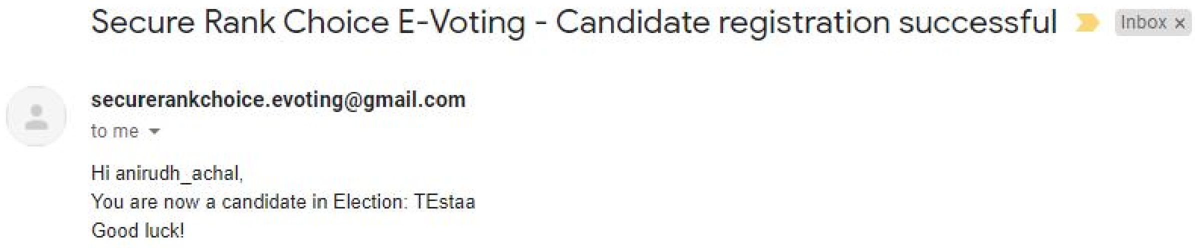
Student Council Head
Registered Voters: 3

Engineer Coordinator
Registered Voters: 3

NITK Football Team Captain
Registered Voters: 3

Cultural Secretary NITK
Registered Voters: 7

If you stand as a candidate, you will receive the mail

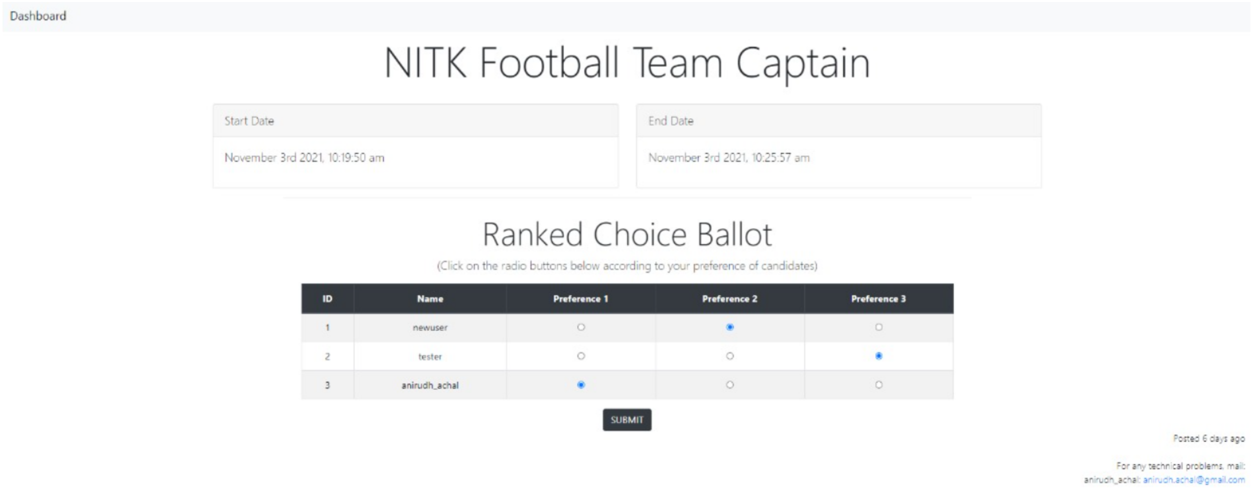


Secure Rank Choice E-Voting - Candidate registration successful

securerankchoice.evoting@gmail.com
to me

Hi anirudh_achal,
You are now a candidate in Election: TEstaa
Good luck!

As it is a ranked choice system, these voting options will be present, one user cannot vote more than once and an error message will be prompted if done so.



Dashboard

NITK Football Team Captain

Start Date	End Date
November 3rd 2021, 10:19:50 am	November 3rd 2021, 10:25:57 am

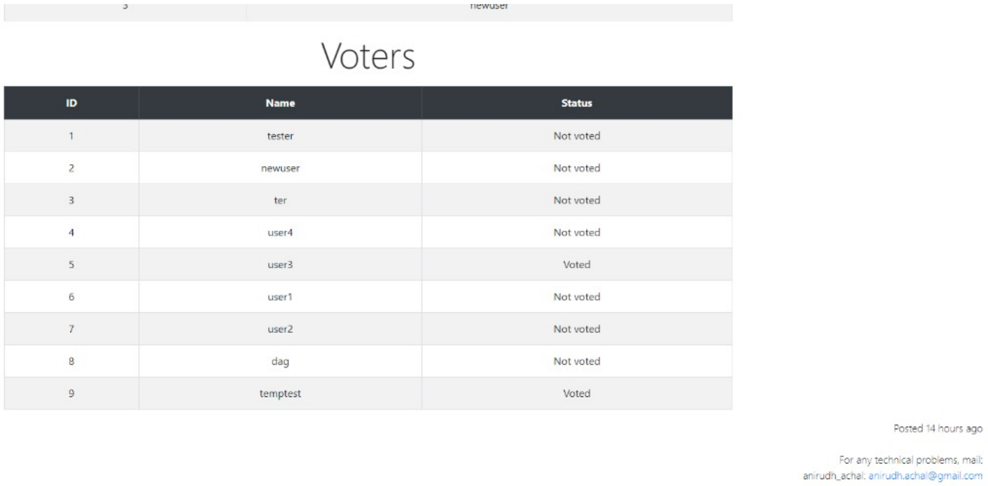
Ranked Choice Ballot

(Click on the radio buttons below according to your preference of candidates)

ID	Name	Preference 1	Preference 2	Preference 3
1	newuser	○	●	○
2	tester	○	○	●
3	anirudh_achal	●	○	○

Posted 6 days ago

For any technical problems, mail:
anirudh_achal: anirudh.achal@gmail.com



3 newuser

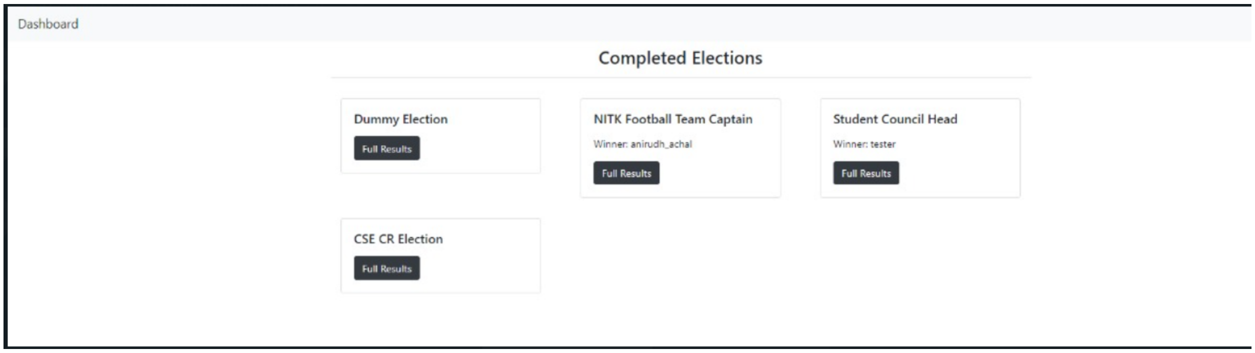
Voters

ID	Name	Status
1	tester	Not voted
2	newuser	Not voted
3	ter	Not voted
4	user4	Not voted
5	user3	Voted
6	user1	Not voted
7	user2	Not voted
8	dag	Not voted
9	temptest	Voted

Posted 14 hours ago

For any technical problems, mail:
anirudh_achal: anirudh.achal@gmail.com

After the elections have been completed, Results will be displayed in “Completed elections”



Dashboard

Completed Elections

Dummy Election

NITK Football Team Captain

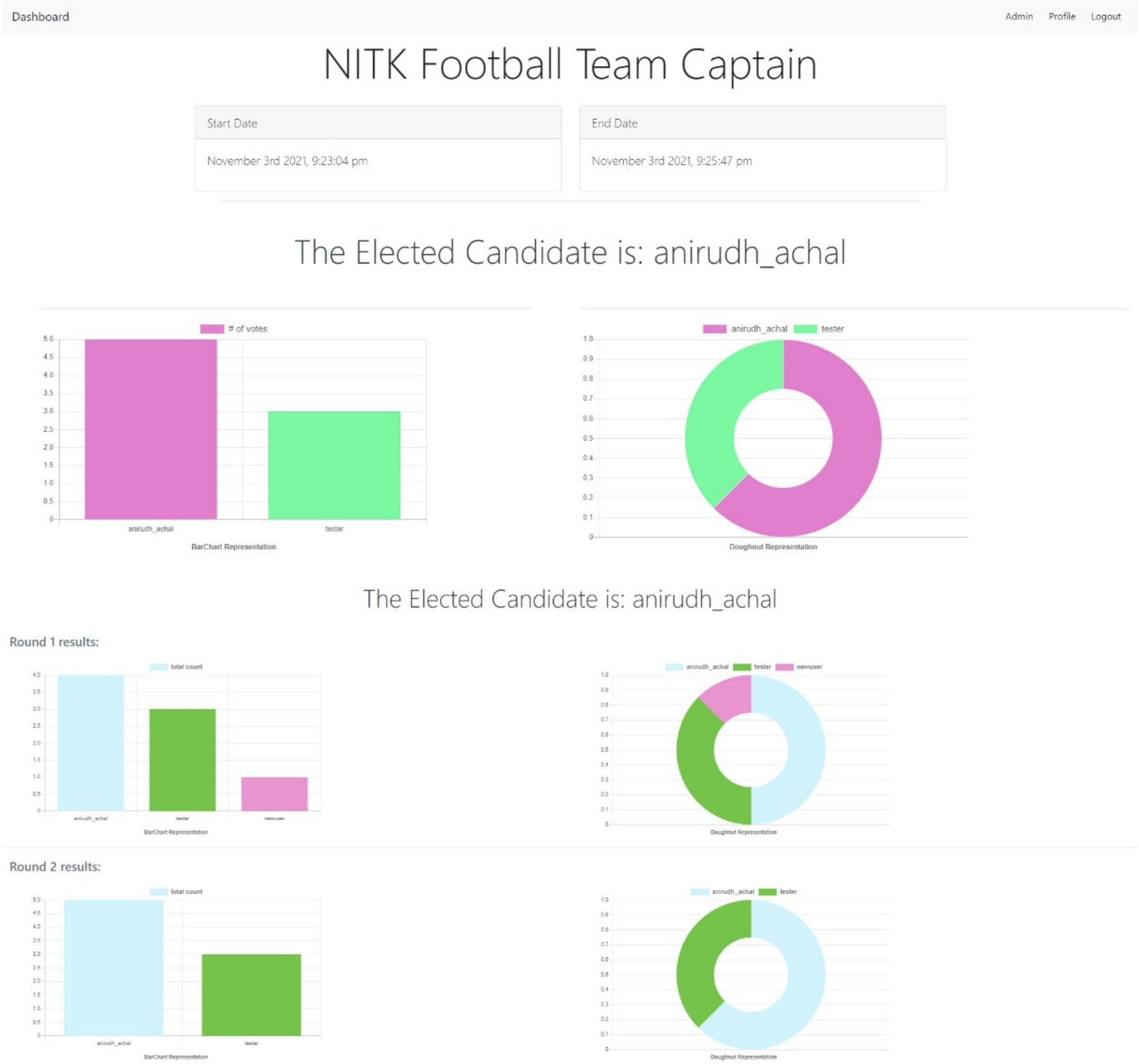
Winner: anirudh_achal

Student Council Head

Winner: tester

CSE CR Election

Clicking on “Full results”, will show detailed distribution of votes.
Since we ranked choice single transferable vote, results will offer preferences in each round of the election.



NITK Football Team Captain

Start Date: November 3rd 2021, 9:23:04 pm

End Date: November 3rd 2021, 9:25:47 pm

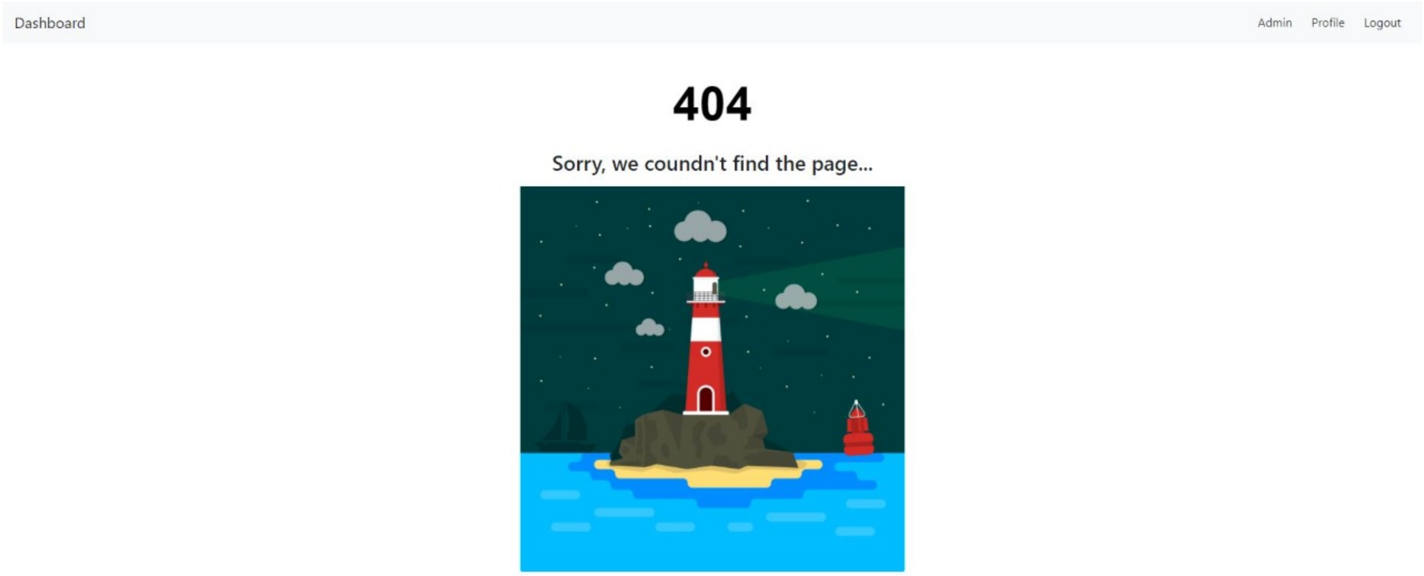
The Elected Candidate is: anirudh_achal

The Elected Candidate is: anirudh_achal

Round 1 results:

Round 2 results:

If there is any error in a user request, or some user is not available and we try finding them, then an error page will appear.

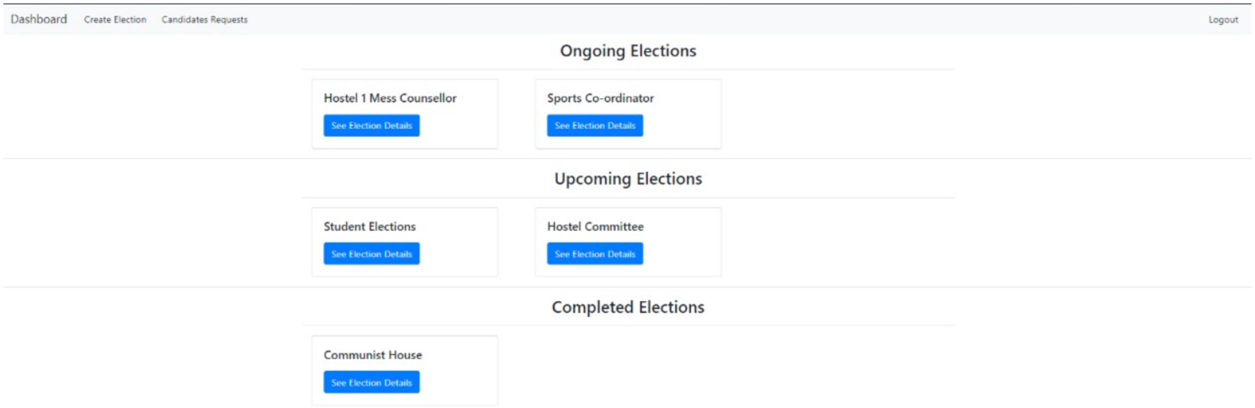


404

Sorry, we coundn't find the page...

ADMIN:

For the dashboard of admin, It'll be more elaborate. Admin will need to name the elections as authorize certain users who shall be able to participate in said elections (by voting).



Ongoing Elections

Hostel 1 Mess Counsellor

Sports Co-ordinator

Upcoming Elections

Student Elections

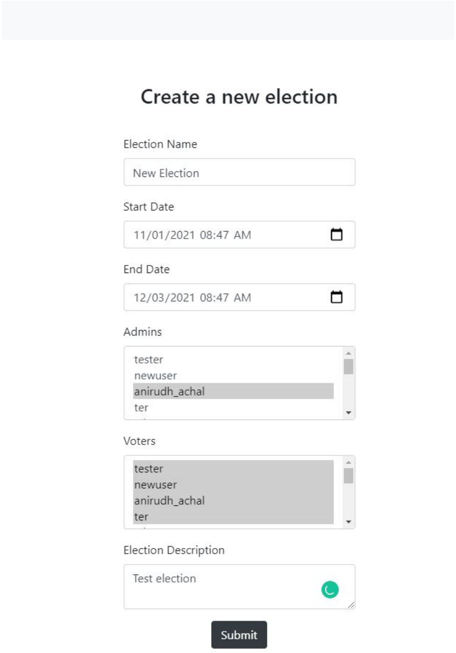
Hostel Committee

Completed Elections

Communist House

Admin can create an election with the valid information. Due to security features we have implemented, no admin can see the actual ballots, hence it will be a secure election with the colloquial "secret ballot".

Admin will also need to sate a timeline for the vote along with a short description of what the election is for.



Create a new election

Election Name: New Election

Start Date: 11/01/2021 08:47 AM

End Date: 12/03/2021 08:47 AM

Admins: tester, newuser, anirudh_achal, ter

Voters: tester, newuser, anirudh_achal, ter

Election Description: Test election

7.Testing and maintenance

7.1.Bugs spotted

During the implementation of our project, there were several bugs found and they are listed down below.

BUGS:

- **BUG:** Users could select more than one preference for a candidate.

  FIX: Radio buttons were used, so user can select only one preference and choosing a new preference would clear the old preference

  STATUS: FIXED.

- **BUG:** Error in result calculation in ranked choice voting.

  FIX: After a candidate is eliminated, their votes for the next round will be 0. So while choosing the candidate in the next round, you need to skip over the already eliminated contestants who will otherwise be chosen as the lowest number.

  STATUS: FIXED.

- **BUG:** Website crashes when no one votes in the election

  FIX: Adding a null condition where if no one votes, result will be shown as null

  STATUS: FIXED.

- **BUG:** Still able to vote after elections ended

  FIX: Set a time check for election, error statement would be shown “The election has ended”

  STATUS: FIXED.

- **BUG:** Users would be able to go to the dashboard without authenticating their account.

  FIX: If the account is not authenticated, User would be redirected to the log-in page.

  STATUS: FIXED.

- **BUG:** Backend fails if there are no winners for the election.

  FIX: Add an if condition to ensure we are only declaring a winner if there is a winner present.

STATUS: Fixed

7.2.Unit tests

The project required two different kinds of test cases,

7.2.1.Election model: To make sure you can get current and upcoming elections only if you are logged in otherwise you should get an unauthorized response from the server.

7.2.2.User Model: We have to make sure the users can get an access token only if their account is active.

UNIT TESTS	**TEST CASES**	**CASES PASSED**	**STATUS**
Election model	4	4	SUCCESS
User model	4	4	SUCCESS

7.3.Importance of maintenance

Creating a brand new piece of software and releasing it is an exciting step for any team. A lot of effort goes into creating your project and its presentation including the actual building and coding, testing models, documentation, and more. However, any great project must be able to adapt to the times.

7.4.Maintenance Attributes

Hence maintenance is of utmost importance. To assist us in our process, we have divided our maintenance into 4 parts. Each of them are discussed below:

7.4.1.Corrective Maintenance

Corrective software maintenance is the basic or classical form of maintenance. Corrective software maintenance is necessary when something goes wrong in a piece of software including faults and errors. These have a widespread effect on the function of the software in general and therefore must be addressed as swiftly as possible.

- We encountered a few issues regarding the preferential ballot which were resolved with the use of appropriate radio buttons to make sure only and only one candidate may be chosen for a particular rank preference.

- Another issue faced was with the implementation of the single transferable vote scheme. As the method suggests transferring votes after every round, it might happen that a previously eliminated candidate gets votes later on. To mitigate this we have altered the algorithm to ensure this doesn't cause calculation errors.

- Websites had crashed on certain occasions when no ballots were received, this was appropriately through the systematic use of the null condition in the code.

- An issue with the backend was found in case the outcome of the election was not found, this was also fixed.

7.4.2. Preventive Maintenance

Preventive software maintenance may address small issues which at the given time may lack significance but may turn into larger problems in the future. These are called latent faults which need to be detected and corrected to make sure that they won't turn into effective faults.

- Authentication was not perfect since a user could view the dashboard without submitting username and password. Even though this system doesn't change the outcome of any election, this extra layer of security makes the project safe from attacks in the future.

- Adequate error messages were missing, this didn't change the functionality of the project but it is always better to let the user know about what is happening with regards to the elections in his/her organization, i.e. it makes the experience more user friendly and improves the outlook of the webpage.

7.4.3.Perfective Maintenance

Perfective software maintenance aims to adjust software by adding new features as necessary and removing features that are irrelevant or not effective in the given software. We are yet to release this project to the public, hence as of right now, we have made these improvements by self-testing these features and brainstorming ideas.

- After numerous tests from both user and election model some more maintenance was required. The colour scheme was altered to make it better for the user to read.

- An addition of a dark mode has been made, this makes the screen better for use in dimly lit areas.

7.4.4. Adaptive Maintenance

Adaptive software maintenance has to do with the changing technologies as well as policies and rules regarding your software. These include operating system changes, cloud storage, hardware, etc. When these changes are performed, your software must adapt in order to properly meet new requirements and continue to run well. We have a plan for the upscaling of this project but since the long-term effect hasn’t materialized we are sticking to the web-based application.