

A
PROJECT REPORT
ON
ONLINE VOTING SYSTEM
By
HATVANI MITTAL V. (CE-68) (19CEUBS126)
NINAMA HEMANGINI S. (CE-78) (17CEUTG004)

B.Tech CE Semester-IV

Subject: Software Engineering Practice and Project

Guided by:
Prof. Pinkal Chauhan
&
Prof.Brijesh S. Bhatt & Prof. Jigar Pandya



Faculty of Technology

Department of Computer Engineering

Dharmsinh Desai University

Contents:

1. Abstract	4
2. Introduction.....	5
3. Software Requirement Specifications	7
4. Design	
I) Use Case diagram.....	11
II) Sequence diagram.....	12
III) Activity diagram	14
IV) Class diagram.....	16
V) Structure chart.....	17
VI) Data flow diagram	18
5. Implementation Detail.....	20
6. Testing.....	25
7. Work flow.....	27
8. Conclusion	33
9. Limitation	33
10. Bibliography	33

1. Abstarct

At the end of any political term, millions of voters are called upon to cast their votes for their next political representatives.

Unfortunately, many eligible voters will not be able to reach a polling station during the election.

Some are living abroad or are deployed in the military. Some have disabilities and face inaccessible polling locations. Some simply don't have the means to take the time to vote, whether it be due to a job, to travels, or to living far away from a voting center.

But if citizens could vote from anywhere, and at any time, then these hurdles wouldn't exist.

This is exactly where online voting comes in. It gives voters the possibility to easily and comfortably fill out and cast a ballot using their own personal computer or smartphone, allowing them to vote from virtually any location with Internet access.

2.Introduction

2.1 Brief Introduction

Electronic voting is often seen as a tool for making the electoral process more efficient and for increasing trust in its management. Properly implemented, e-voting solutions can increase the security of the ballot, speed up the processing of results and make voting easier. However, the challenges are considerable. If not carefully planned and designed, e-voting can undermine the confidence in the whole electoral process. This policy paper outlines contextual factors that can influence the success of e-voting solutions and highlights the importance of taking these fully into account before choosing to introduce new voting technologies.

2.2 Tools/Techologies Used

Technologies:

- o Django
- o Python
- o MySQL
- o Bootstrap
- o JavaScript
- o HTML

Tools:

- o Git
- o Visual Studio Code
- o WonderShare Edrawmax IDE (for design document)

Platform:

Local development server

3. Software Requirement Specification

Online voting system:-

3.1 System Functional Requirements

R.1: MANAGE HOMEPAGE

R.1.1:DISPLAY LINK OF OTHER PAGES

Description: That is saw the different buttons for go to that page directly

Ip : click on that button

Op: reach to that page

R.2:MANAGE REGISTRATION

Description: if we are new user or candidate we have to register in that page.

R.2.1:ENTER DETAILS OF VOTERS

Ip: entre the details of user

Op: successfully register message

R.2.2:ENTER DETAILS OF CANDIDATE

Ip: entre the details of candidate

Op: successfully register message

R.3:MANAGE USER LOGIN

Description: login before voting or participate in election

R.3.1:SAVE DATABASE

Description: save the entire data which is entre by user in login page

R.3.2:GENERATED PASSWORD

Ip: new password creation

Op: password create successfully

R.3.3:FORGOT PASSWORD

Ip: old password forgot than create new one

Op: password create successfully

R.3.4:LOGOUT

Description: if all the process is done or you want out of the page used logout button.

R.4:MANAGE ADMIN PANEL

Description: manage the all details about election

R.4.1:ADD NEW ELECTION

Description: if you want add new election in that process it is used

R.4.2:VERIFYING USERS

Ip: entre the data for verification

Op: your data is verified.

R.4.3:GENERATING ID FOR USER

Description: it is use when user entre details than it is creat unique id for that user.

Ip: entre the password for id

Op: id is created

R.5:ELECTION

R.5.1: LIST OF ON GOING ELECTION

Description: it is describe the data of which election are going at that time and give that list.

R.5.2:VOTING (SELECT CANDIDATE)

Description: vote the candidate which you want to win election

Ip: select any button for vote.

Op: you have successfully voted.

R.6:MANAGE RESULT

R.6.1:GENERATING RESULT

Description: counts the votes and create result of election and who is winner.

R.6.2:DECLARE WINNER

Description: it is saws that who is winner base on vote counting.

R.7:CHATBOX

Description: if you have give any suggestion or anything else give here.

R.7.1:FEEDBACK

Ip: give your feedback about the election

Op: your feedback is taken thank you for giving.

R.7.2:WRITE ANY QUERY

Description: if you have any question or not understood than you write here for solve that problem.

Ip: write a query

Op: thank you for asking we will give reply soon

3.2 Other Nonfunctional Requirements

Convenience: The system shall allow the voters to cast their votes quickly, in one session, and should not require many special skills or intimidate the voter (to ensure Equality of Access to Voters).

User-Interface: The system shall provide an easy-to-use user-interface. Also, it shall not disadvantage any candidate while displaying the choices (e.g., by requiring the user to scroll down to see the last few choices).

Transparency: Voters should be able to possess a general knowledge and understanding of the voting process.

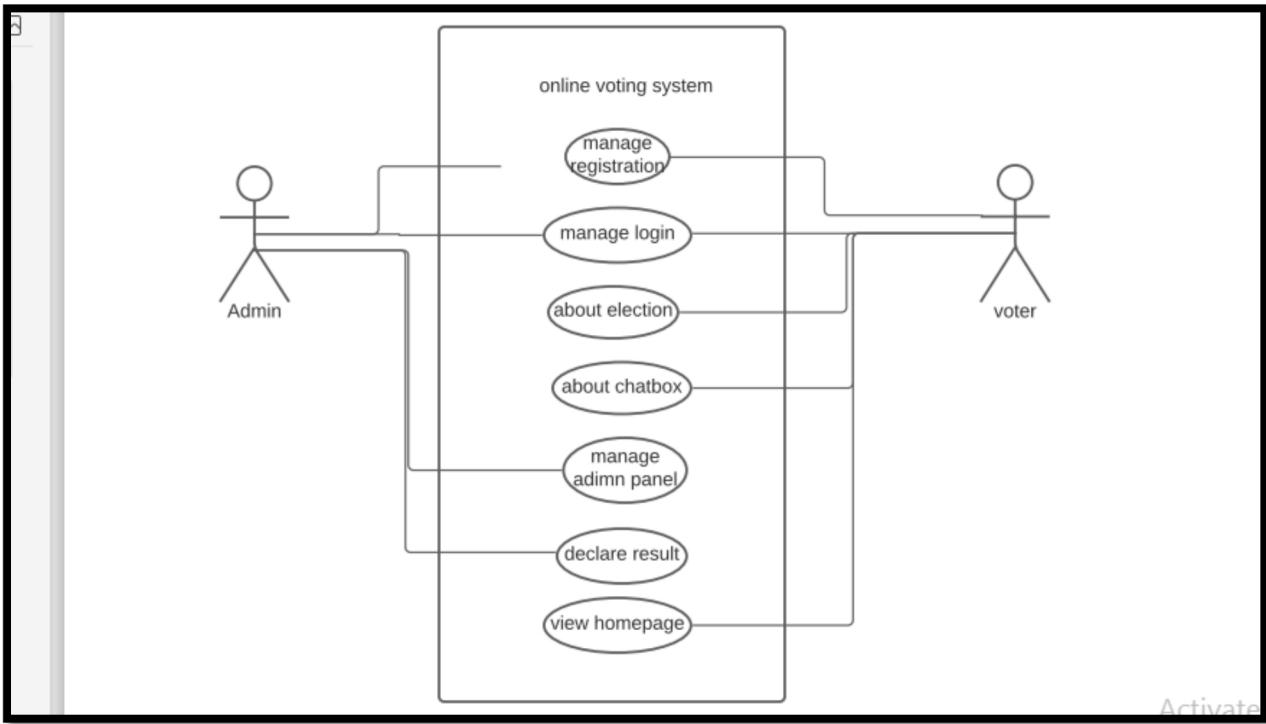
Flexibility: The system shall be flexible in that it allows a variety of ballot question formats including open-ended questions (e.g. Write-in candidates and survey questions).

Support for Disabled Voters: The system shall cater to the needs of physically challenged voters (e.g. blind voters).

Accuracy: The system shall record and count all the votes and shall do so correctly. 8. **Eligibility:** Only authorized voters, who are registered, should be able to vote. 9. **Uniqueness:** No voter should be able to vote more than once.

4.Design

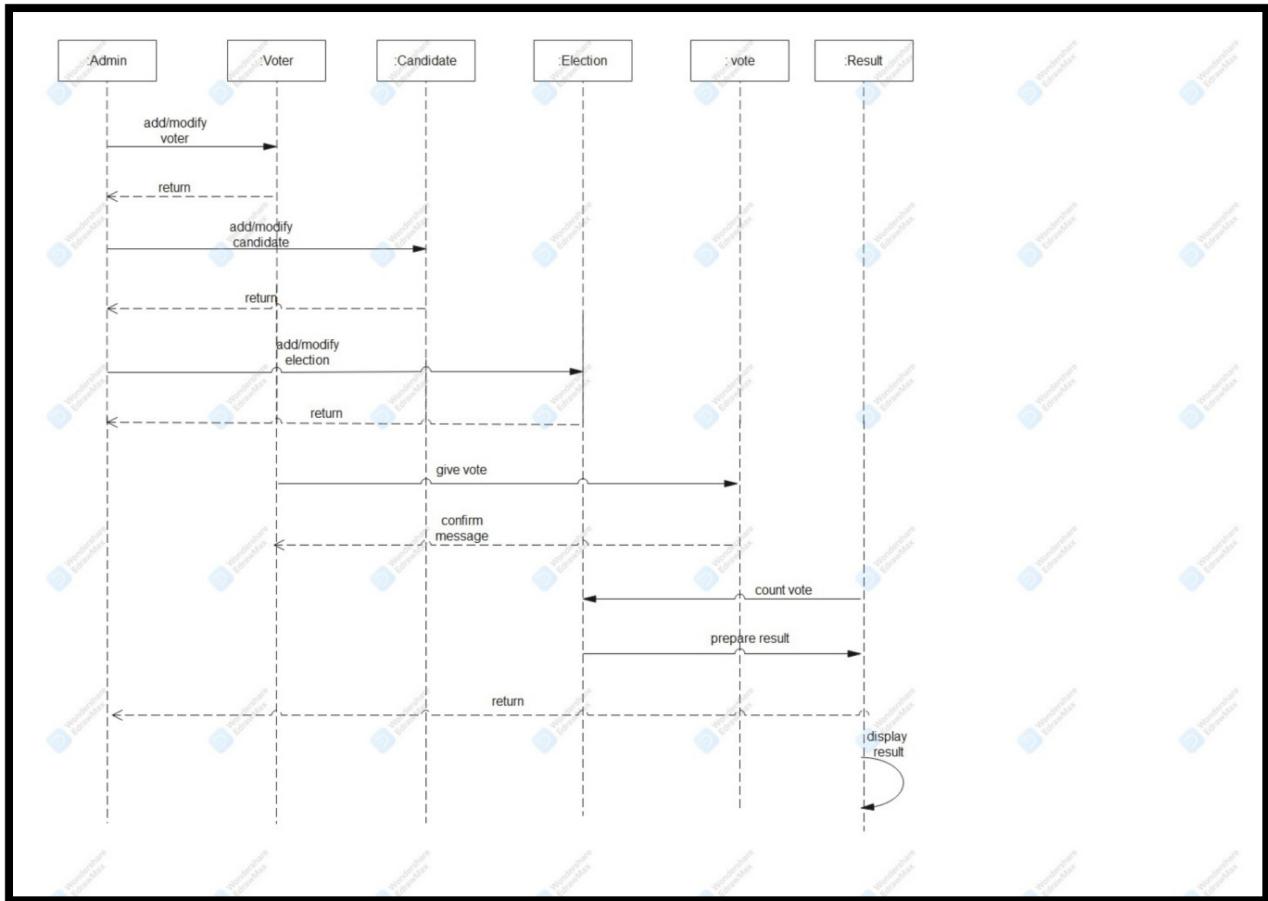
4.1 Use Case Diagram



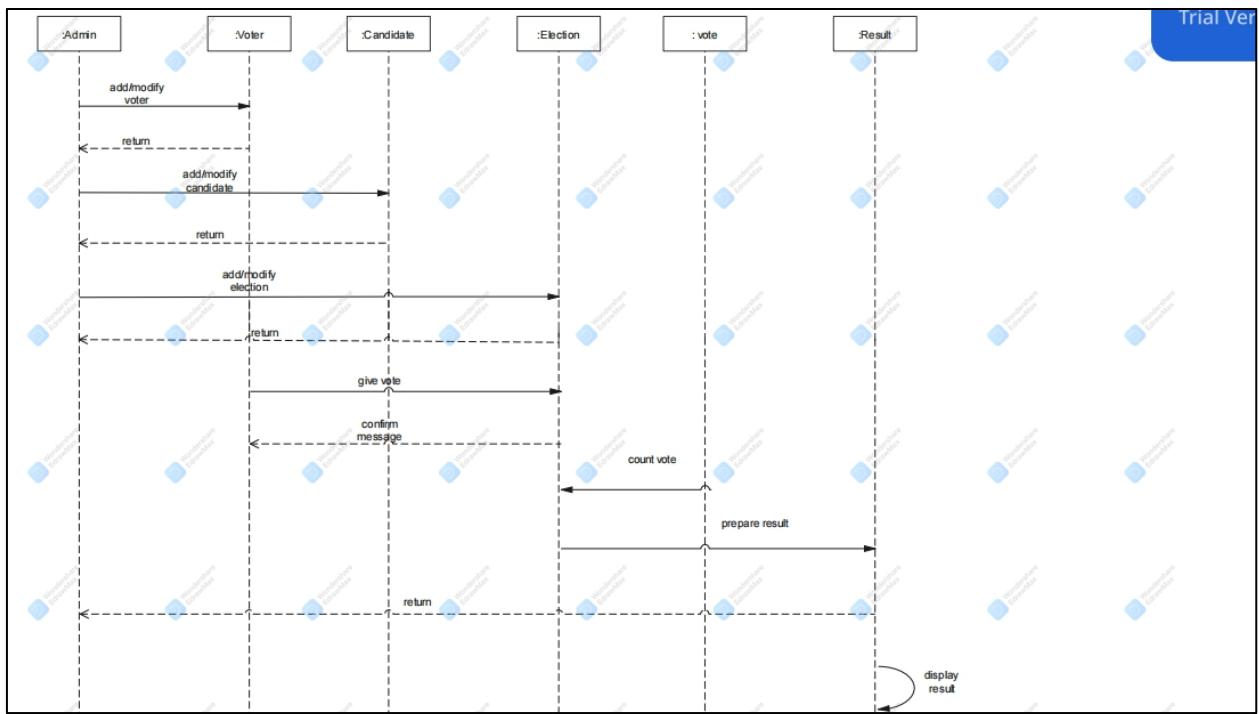
Use case diagram of user

4.2 Sequence Diagram

1)

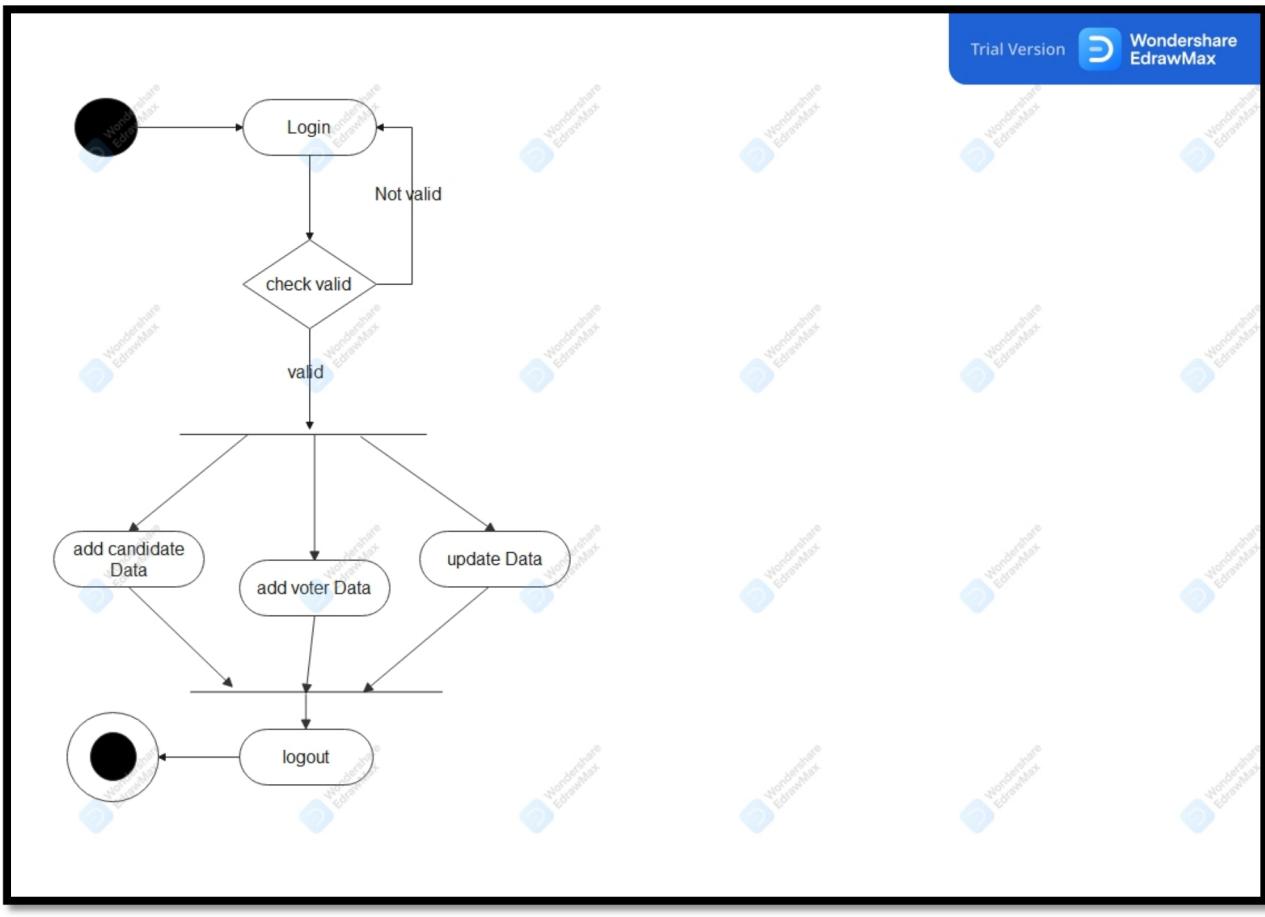


2)

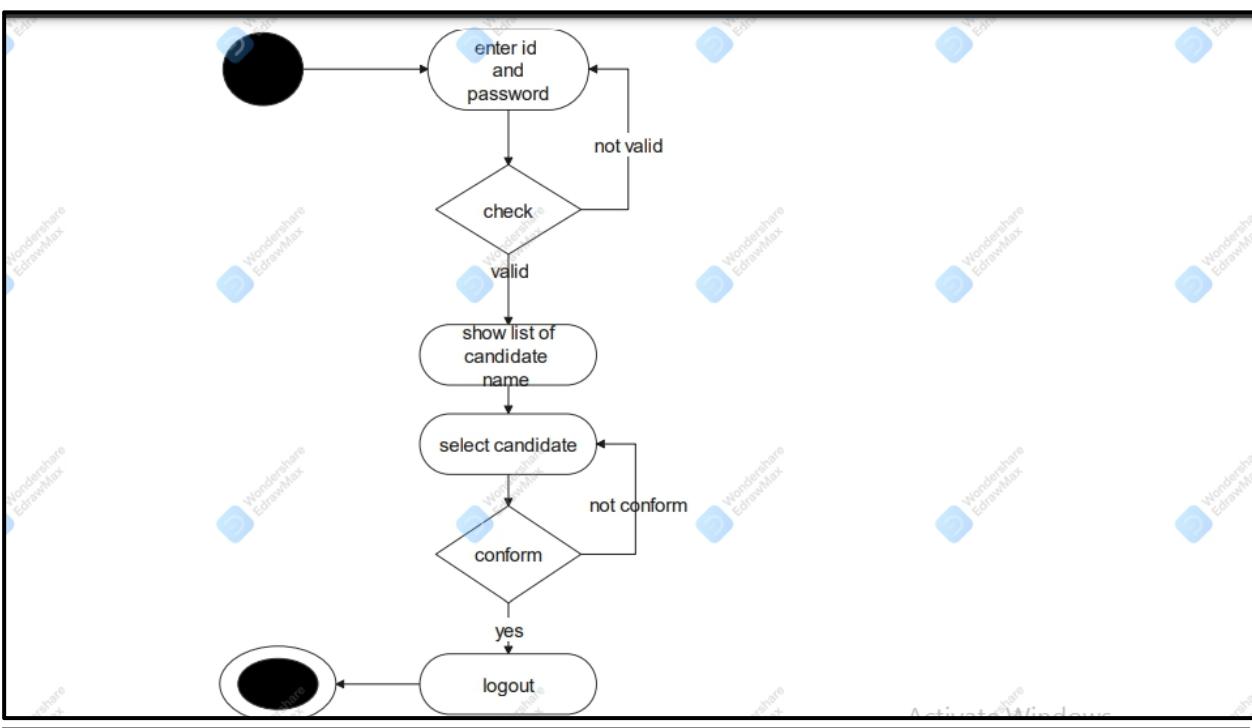


4.3 Activity Diagram

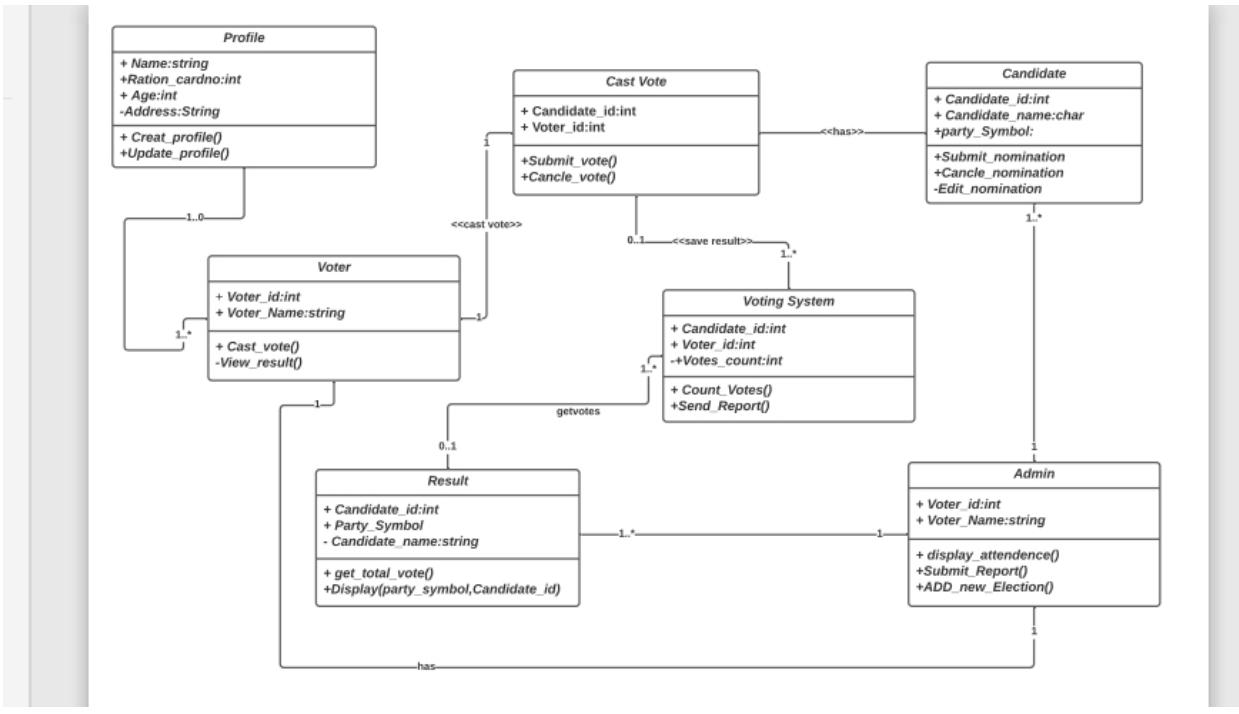
1)



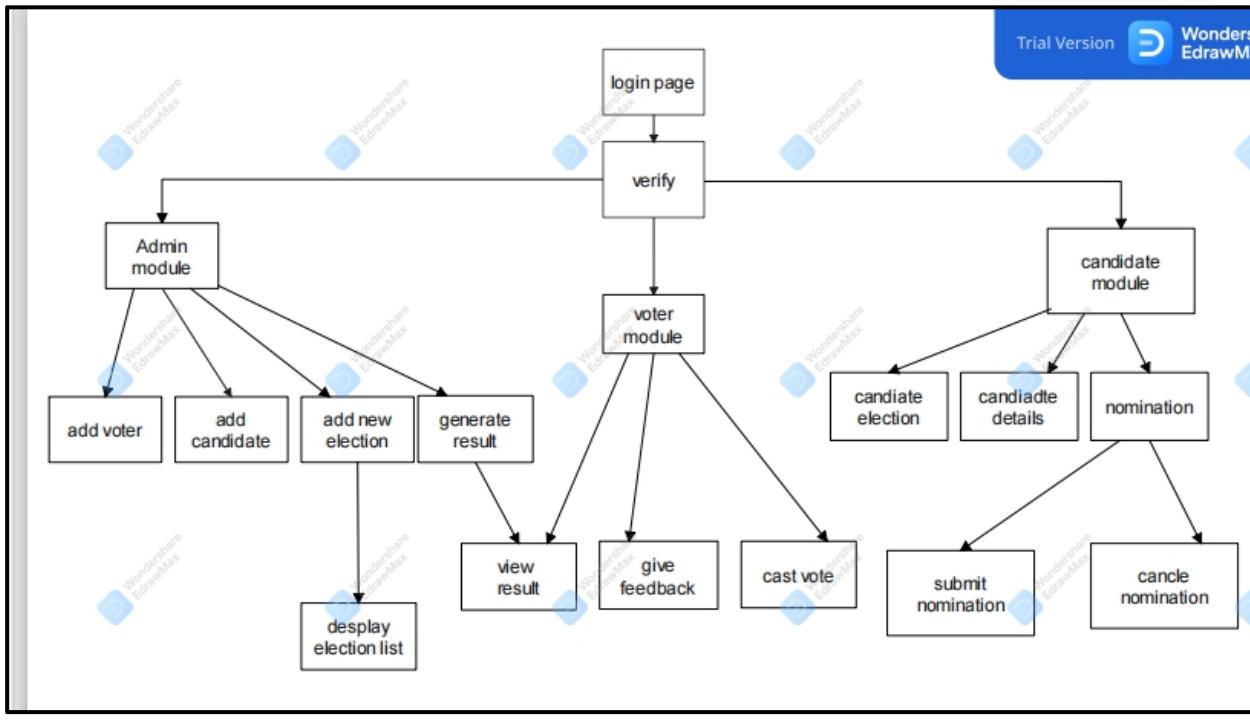
2)



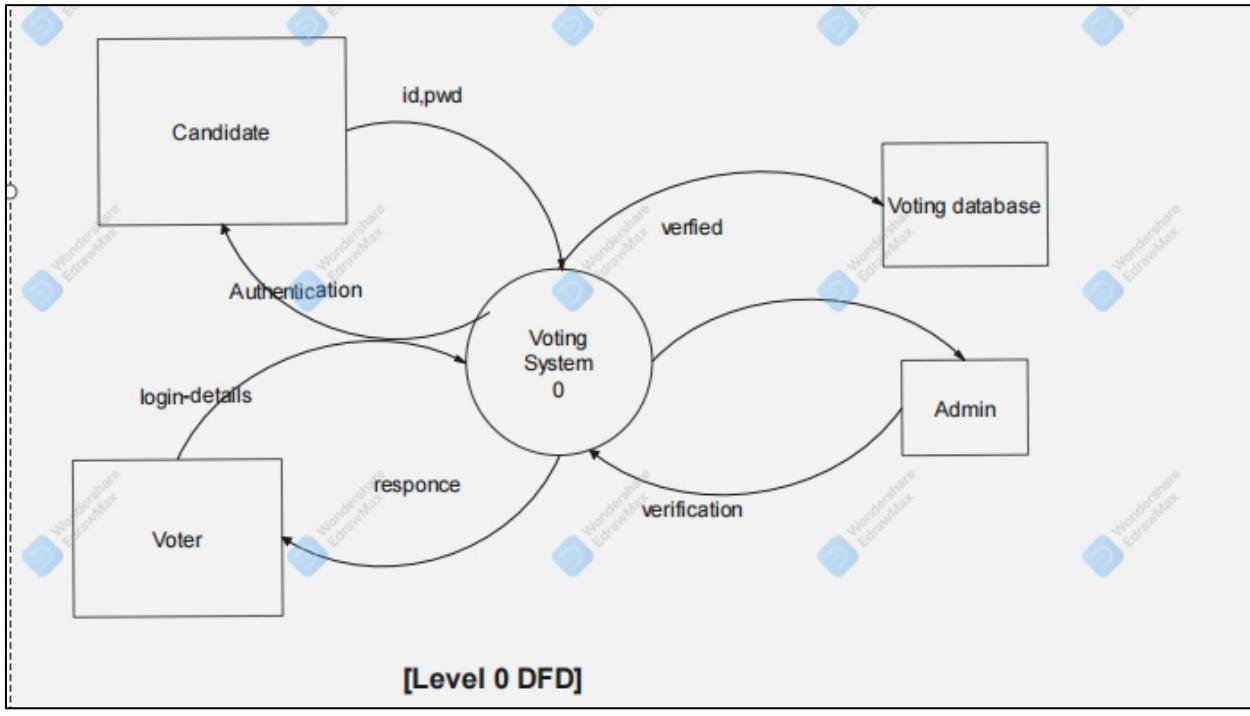
4.4 Class Diagram :

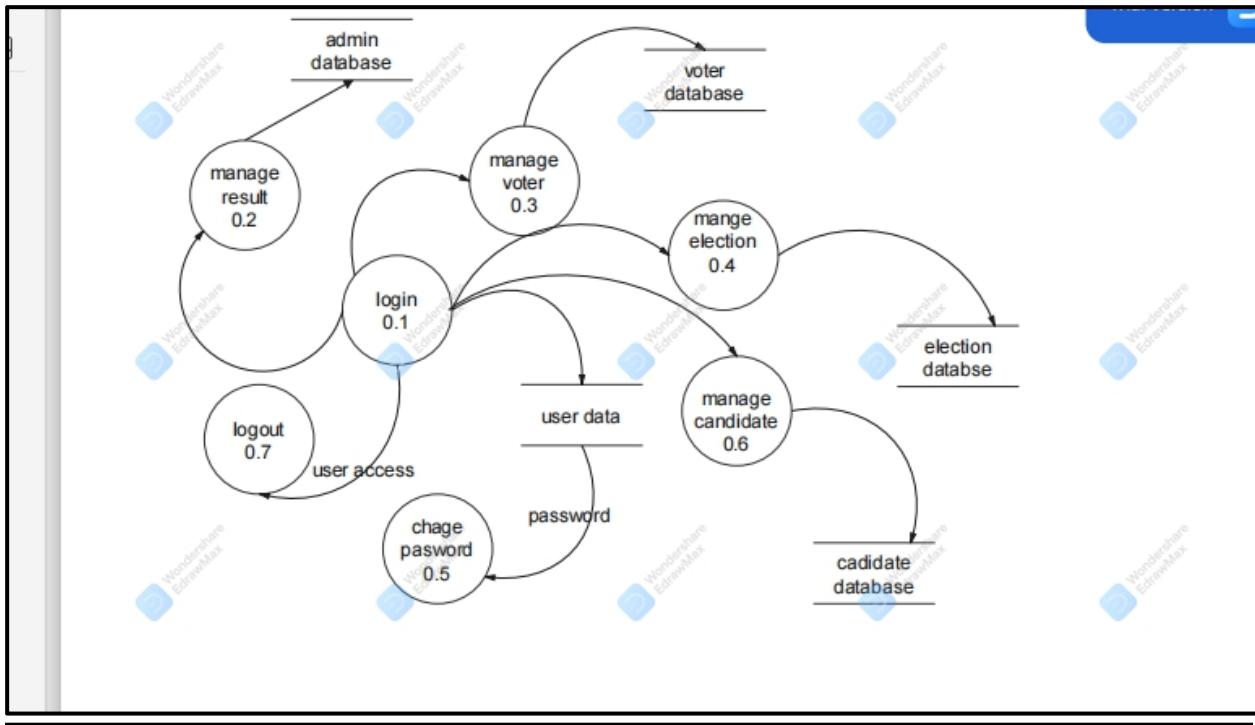


4.5 Structure chart:

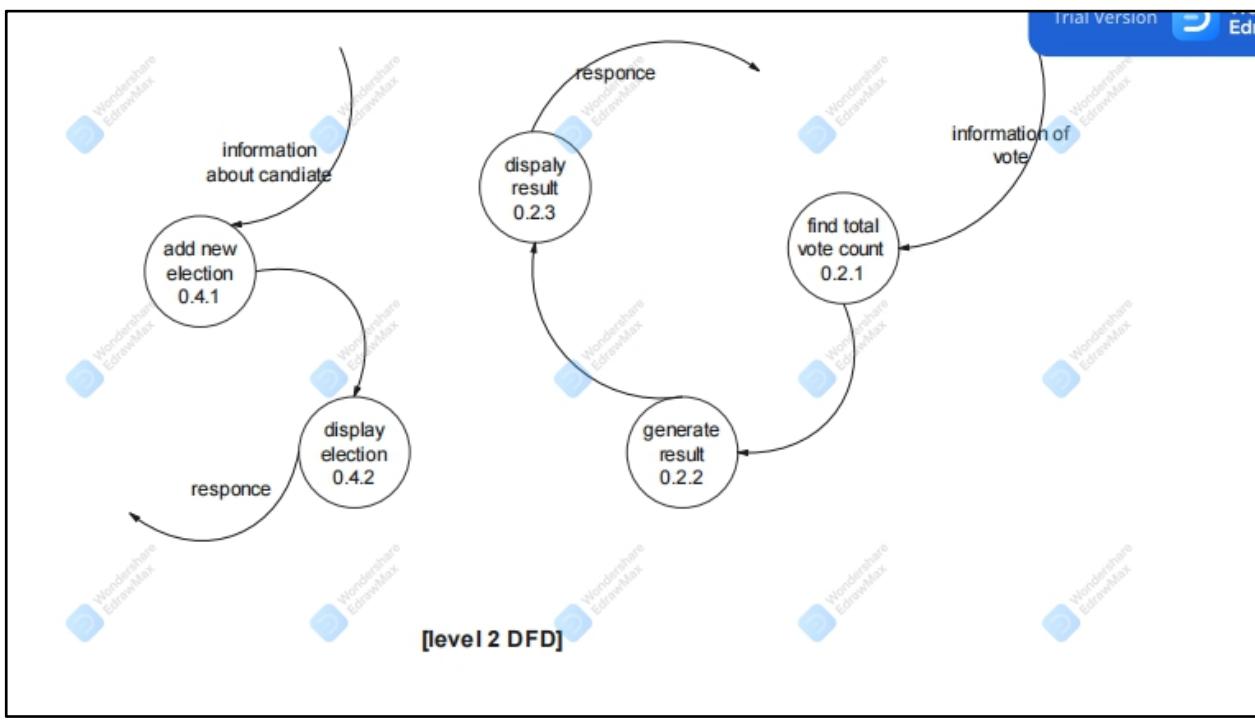


4.6 Data model :





level-1



5.Implementation Details

5.1 Description of Modules :-

1. Register User Module

Basic information of user is taken by system and stored in database.

2. Login User Module

Users are able to login themselves. System logs user in, then and only then user can use other functionalities of system.

3. Vote Module

User are after login they can see voting position and see the all participated candidate and give them vote.

4. Result Module

User are able to see the result before or after voting.

5. Change Password Module

User can change their password by entre the old password.

6.Update Profile Module

User are able to change their profile after login in that they can change username,first name,last name,email.

7.Logout Module

User also logout with their profile.

8. Dashboard Module

In the dash board page user go to voting and also go to the result .

```
views.py  * tests.py  * forms.py  * app.py  * admin.py  * views.py  * bin.py  * README
views.py > registrationView
def registrationView(request):
    return render(request, "poll/home.html")

def registrationView(request):
    if request.method == "POST":
        form = RegistrationForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            if cd['password'] == cd['confirm_password']:
                obj = form.save(commit=False)
                obj.set_password(obj.password)
                obj.save()
                messages.success(request, 'You have been registered.')
                return redirect('home')
            else:
                return render(request, "poll/registration.html", {'form':form, 'note':'password must match'})
        else:
            form = RegistrationForm()

    return render(request, "poll/registration.html", {'form':form})

def loginView(request):
    if request.method == "POST":
        usern = request.POST.get('username')
        passw = request.POST.get('password')
        user = authenticate(request, username=usern, password=passw)
        if user is not None:
            login(request, user)
            return redirect('dashboard')
        else:
            messages.success(request, 'Invalid username or password!')
            return render(request, "poll/login.html")
    else:
        return render(request, "poll/login.html")
```

Registration

```
def loginView(request):
    if request.method == "POST":
        usern = request.POST.get('username')
        passw = request.POST.get('password')
        user = authenticate(request, username=usern, password=passw)
        if user is not None:
            login(request, user)
            return redirect('dashboard')
        else:
            messages.success(request, 'Invalid username or password!')
            return render(request, "poll/login.html")
    else:
        return render(request, "poll/login.html")
```

>Login required

login

```

@login_required
def candidateView(request, pos):
    obj = get_object_or_404(Position, pk = pos)
    if request.method == "POST":

        temp = ControlVote.objects.get_or_create(user=request.user, position=obj)[0]

        if temp.status == False:
            temp2 = Candidate.objects.get(pk=request.POST.get(obj.title))
            temp2.total_vote += 1
            temp2.save()
            temp.status = True
            temp.save()
            return HttpResponseRedirect('/position/')
        else:
            messages.success(request, 'you have already been voted this position.')
            return render(request, 'poll/candidate.html', {'obj':obj})
    else:
        return render(request, 'poll/candidate.html', {'obj':obj})

```

Candidate voting

```

@login_required
def changePasswordView(request):
    if request.method == "POST":
        form = PasswordChangeForm(user=request.user, data=request.POST)
        if form.is_valid():
            form.save()
            update_session_auth_hash(request, form.user)
            return redirect('dashboard')
    else:
        form = PasswordChangeForm(user=request.user)

    return render(request, "poll/password.html", {'form':form})

```

`@login_required`

Change password

```
7 @login_required
8 def editProfileView(request):
9     if request.method == "POST":
10         form = ChangeForm(request.POST, instance=request.user)
11         if form.is_valid():
12             form.save()
13             return redirect('dashboard')
14     else:
15         form = ChangeForm(instance=request.user)
16     return render(request, "poll/edit_profile.html", {'form':form})
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Edit profile

```
    messages.success(request, "you have already been voted in")
    return render(request, 'poll/candidate.html', {'obj':obj})
else:
    return render(request, 'poll/candidate.html', {'obj':obj})

@login_required
def resultView(request):
    obj = Candidate.objects.all().order_by('position', '-total_vote')
    return render(request, "poll/result.html", {'obj':obj})

@login_required
def candidateDetailView(request, id):
    obj = get_object_or_404(Candidate, pk=id)
    return render(request, "poll/candidate_detail.html", {'obj':obj})
```

Result and candidate view

6.Testing:

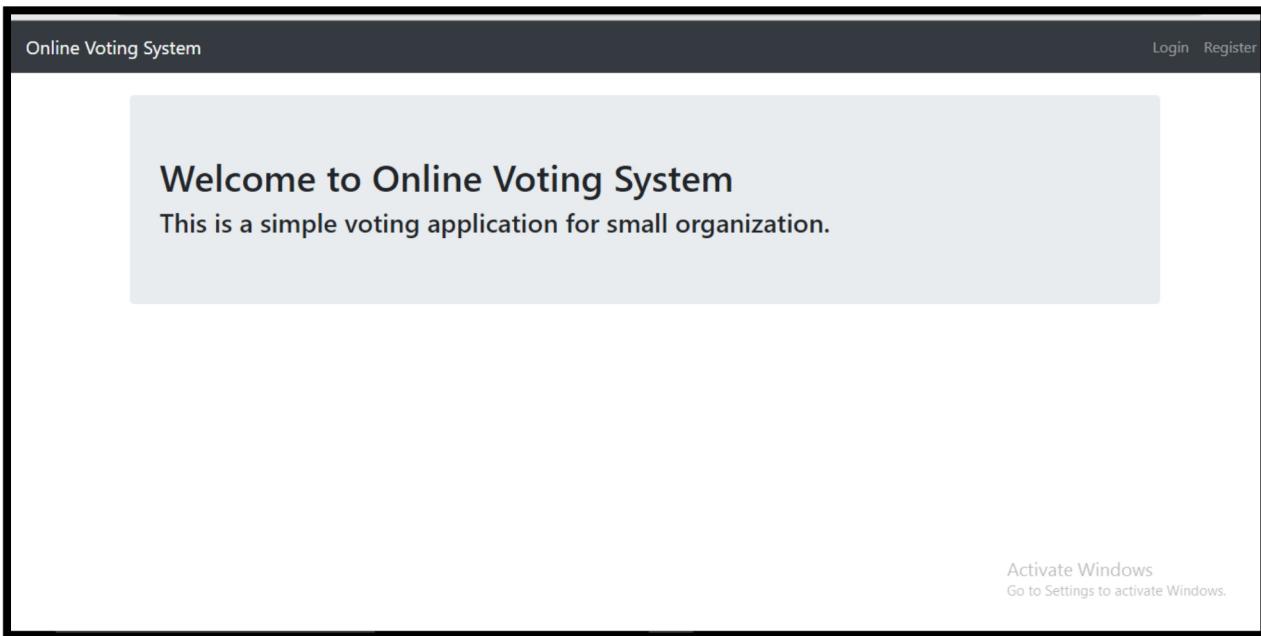
Testing method: manual testing

All test cases executed by us manually according to end user's perspective . our motto to test project is that it ensures matching with requirement document or not

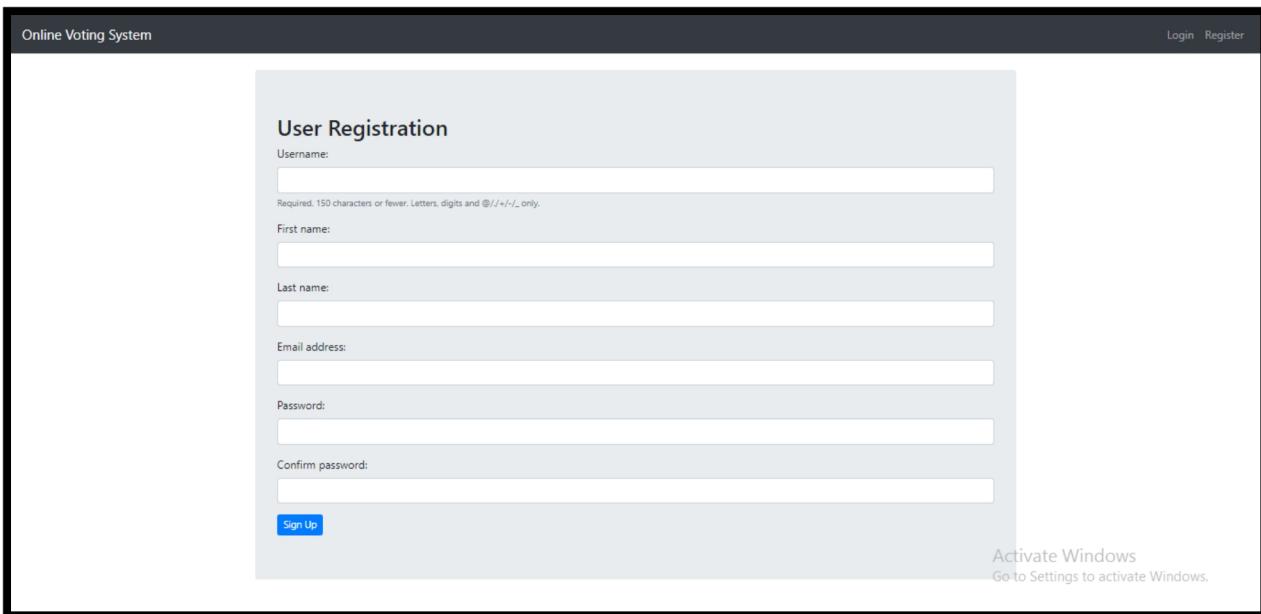
Sr. No.	Test Scenario	Expected Result	Actual Result	Status
1.	Login with incorrect password	User should not able to log in.	Error message Invalid data	Pass
2.	Login with incorrect username	User should not able to login	Error message invalid data	Pass
3.	Login with correct credentials	User should able to login	Redirect to dash bored	Pass

4.	Registration with Correct details	User should register	Successfully registraed	Pass
5.	Without selecting candidate give the vote	User should not able to vote	Please give the vote	Pass
6.	Give the vote to more than one at one position	User should not able to vote	You have already voted	Pass
7.	Username already exit use 2 nd time registration	User should not able to register	Username is already used	Pass

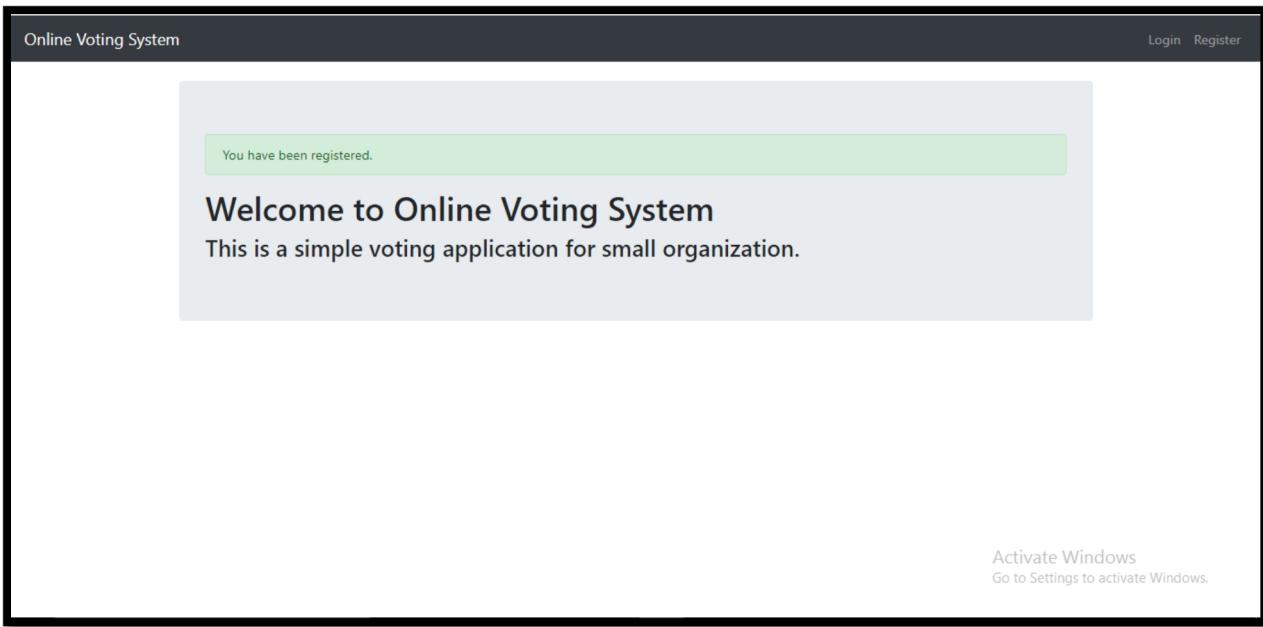
7. Work Flow/Layouts



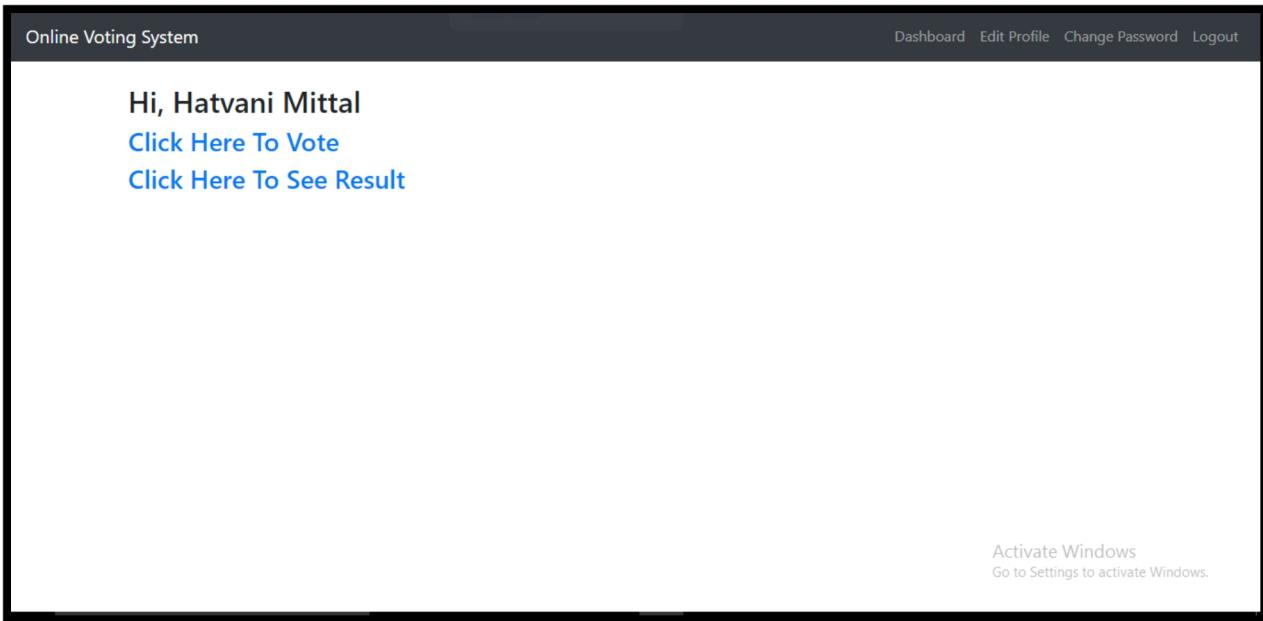
Home Page



Registration page



Registered successfully



Dashboard

Online Voting System

Dashboard Edit Profile Change Password Logout

Update Profile

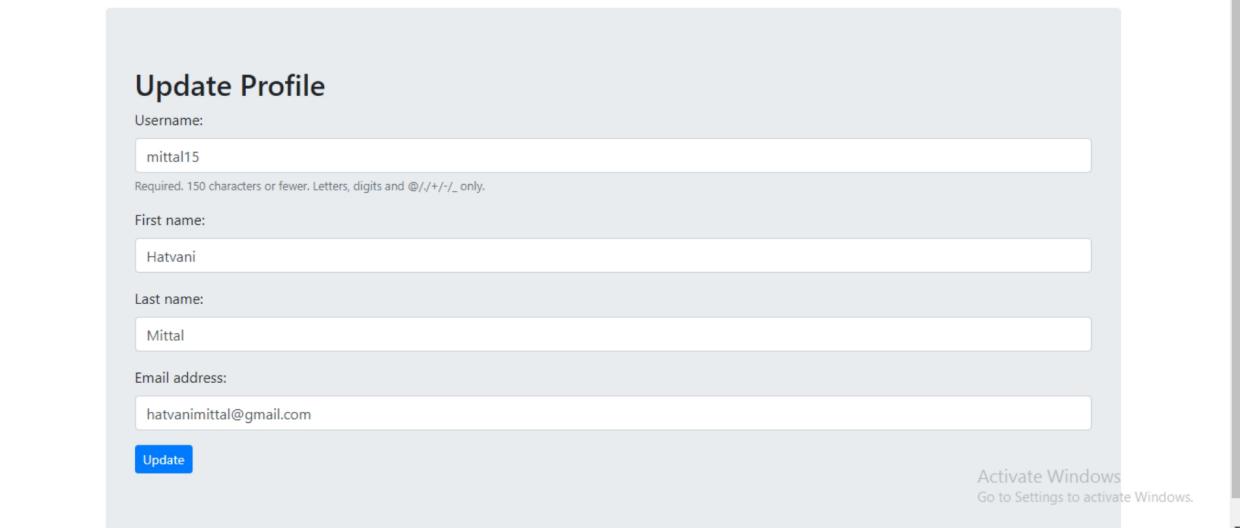
Username: mittal15
Required: 150 characters or fewer. Letters, digits and @/./+/~/_ only.

First name: Hatvani

Last name: Mittal

Email address: hatvanimittal@gmail.com

Activate Windows
Go to Settings to activate Windows.



Update profile

Online Voting System

Dashboard Edit Profile Change Password Logout

Password Change Form

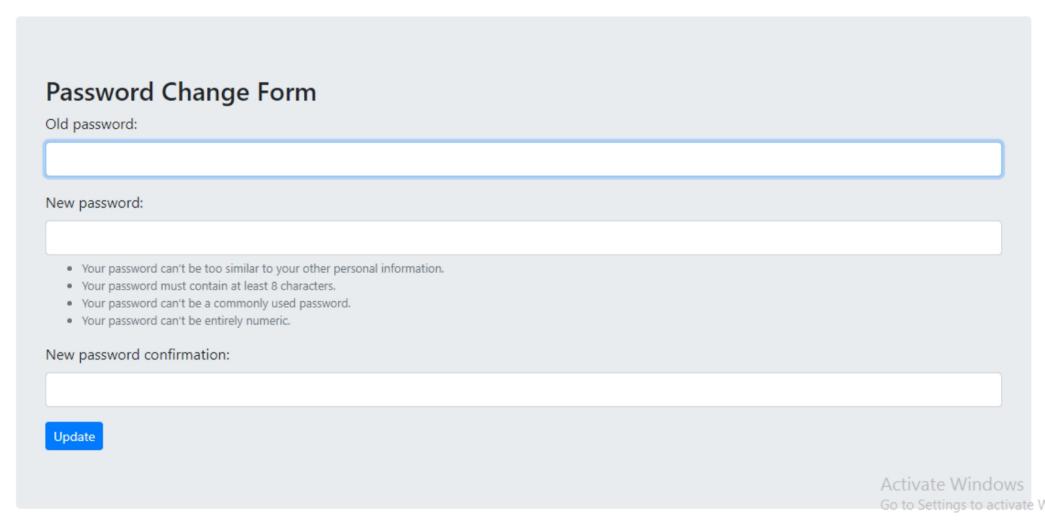
Old password:

New password:

• Your password can't be too similar to your other personal information.
• Your password must contain at least 8 characters.
• Your password can't be a commonly used password.
• Your password can't be entirely numeric.

New password confirmation:

Activate Windows
Go to Settings to activate Windows.



Change password

Available Position For Vote

- President
- Vice President
- Secretary
- Treasurer

Activate Windows
Go to Settings to activate Windows.

Position for vote

Available Candidates of President

- Mittal Hatvani [Detail](#)
- Dharmi Chavda [Detail](#)

[VOTE](#)

[Back to Poll](#)

Activate Windows
Go to Settings to activate Windows.

Candidates for President

Online Voting System

Dashboard Edit Profile Change Password Logout

Available Candidates of President

 Mittal Hatvani [Detail](#)

Please select one of these options.

[VOTE](#)

[Back to Poll](#)

Activate Windows
Go to Settings to activate Windows.

Without selecting candidate give vote (display message)

Online Voting System

Dashboard Edit Profile Change Password Logout



Name: Mittal Hatvani

Position: President

Activate Windows
Go to Settings to activate Windows.

Candidatee Details

you have already been voted this position.

Available Candidates of President

- Mittal Hatvani [Detail](#)
- Dharmi Chavda [Detail](#)

[VOTE](#)[Back to Poll](#)

Activate Windows
Go to Settings to activate Windows.

Already voted Position

Result

Candidate Name	Candidate Position	Total Vote
Mittal Hatvani	President	3
Dharmi Chavda	President	0
Shahid Kapoor	Vice President	2
Geroge Ambani	Vice President	0
Tamjid Sharma	Secretary	1
Arif Bhatia	Secretary	1
Samir Dave	Treasurer	2
Nahian Bhatt	Treasurer	1

Activate Windows
Go to Settings to activate Windows.

Final result

Conclusion

- A ONLINE VOTING SYSTEM can save time spent in voting on the booths.
- Online voting reduces paper work.
- It is NON Expensive and less time consuming.
- 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. there is a DATABASE in which all the names of voter with complete information is stored.

Limitation

- In this user who is above 18 year's register is not necessary we have to implement that only 18+ can register.
- In that system there is problem when people are not aware about online system.
- User must have electronic devise for voting.

Bibliography

References/resources used for developing project:

- <https://nevonprojects.com/online-election-system-project/>
- <https://projectworlds.in/python-django-poll-app-project/>
- www.youtube.com
- www.wikipedia.com

