

voteonline

X +

← → Q CC



VoteOnline

Prashant - Sachin - Michael - Divyesh - Dhruv

Code Crafters

voteonline

Elevetors

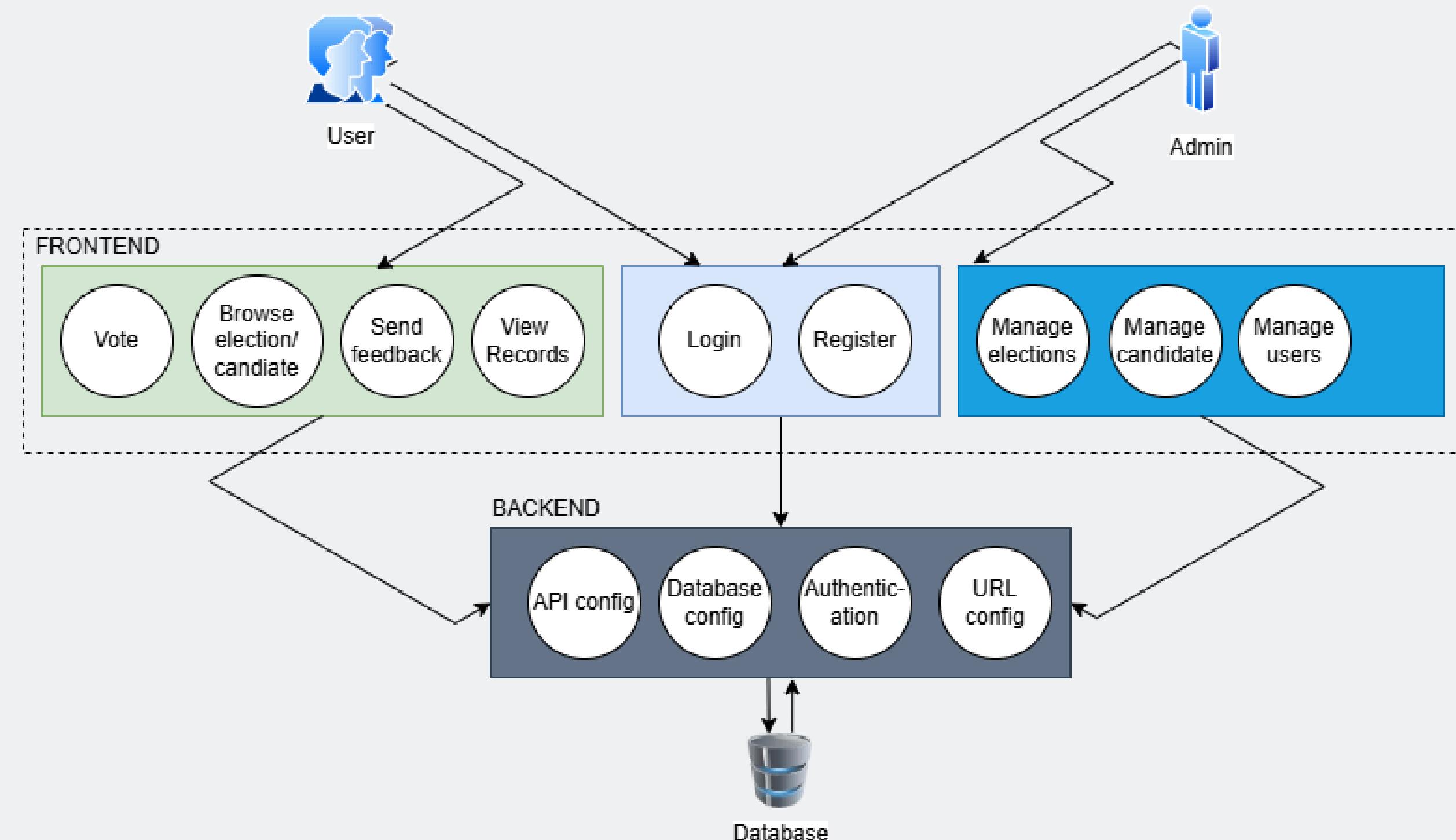
Overview

X

+

← → Q cc/overview

Overview



[voteonline](#)[elevators](#)[overview](#)[component](#)

X

+

[← → Q cc/component-us1](#)

Component of complexity.

voteonline

elevators

overview

component

X

+

← → Q cc/component-us1

User story 1:
As a student , I want to register and login to the system

Requirements:

- User should have WSU email for registration.
- Department should be selected.
- Strong password are only eligible.
- OPT would be sent via email



[voteonline](#)[elevators](#)[overview](#)[component](#)

X

+

[← → Q cc/component-us1](#)

User story 1: Registration

WSU VoteOnline

127.0.0.1:8000/register

Create an Account

gamer.trill22@gmail.com CECS

Prashant Rai

***** *****

Only WSU email accepted."example: abc@wright.edu"
Password can't be too similar to other personal information.
Password must contain at least 8 characters.
Password can't be a commonly used password.
Password can't be entirely numeric.

Register

Already have an account? [Login!](#)



WSU VoteOnline

127.0.0.1:8000/register

Create an Account

Invalid Credentials

gamer.trill22@gmail.com CECS

Prashant Rai

Password Confirm Password

Only WSU email accepted."example: abc@wright.edu"
Password can't be too similar to other personal information.
Password must contain at least 8 characters.
Password can't be a commonly used password.
Password can't be entirely numeric.

Register

Already have an account? [Login!](#)

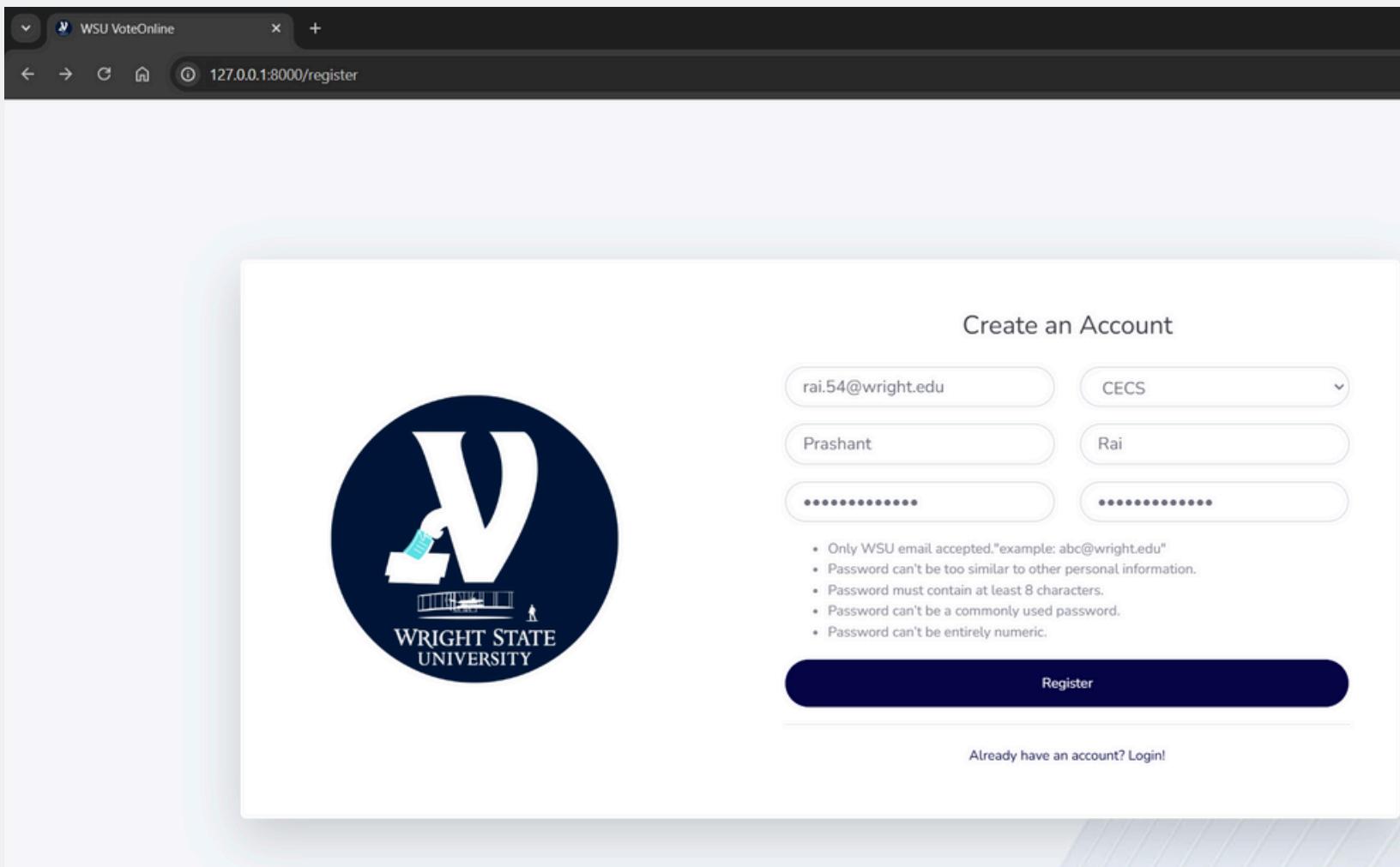
[voteonline](#)[elevators](#)[overview](#)[component](#)

X

+

[← → Q cc/component-us1](#)

User story 1: Registration



WSU VoteOnline

127.0.0.1:8000/register

Create an Account

rai.54@wright.edu
CECS

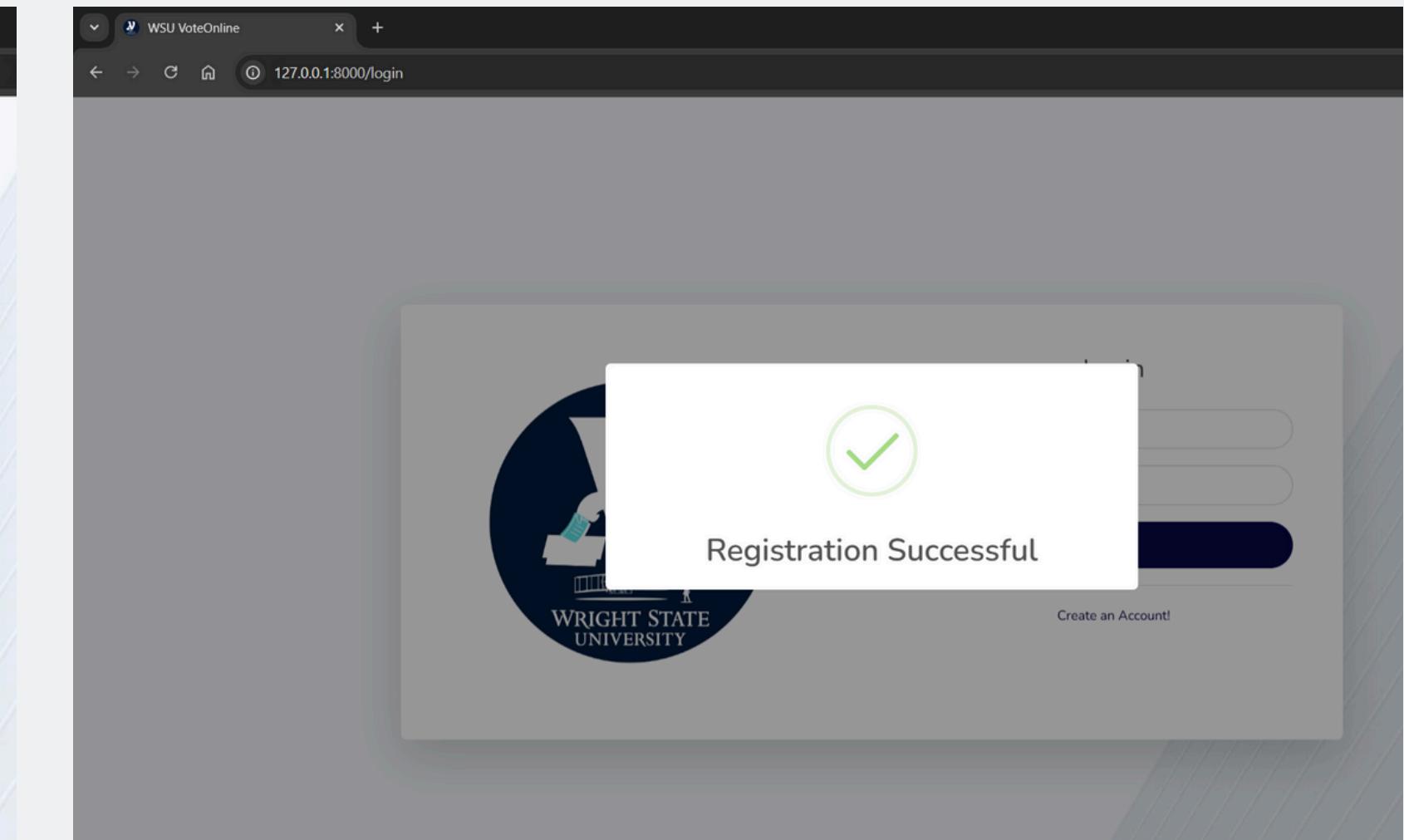
Prashant
Rai

Only WSU email accepted."example: abc@wright.edu"
• Password can't be too similar to other personal information.
• Password must contain at least 8 characters.
• Password can't be a commonly used password.
• Password can't be entirely numeric.

Register

Already have an account? [Login!](#)

The registration form is displayed on a browser window titled "WSU VoteOnline". The URL in the address bar is "127.0.0.1:8000/register". The form is titled "Create an Account". It contains four input fields: email (rai.54@wright.edu), department (CECS), first name (Prashant), and last name (Rai). Below the inputs is a password field with masked input. A list of validation rules is shown: Only WSU email accepted."example: abc@wright.edu", Password can't be too similar to other personal information, Password must contain at least 8 characters, Password can't be a commonly used password, and Password can't be entirely numeric. A large blue "Register" button is at the bottom. At the bottom of the form, there is a link "Already have an account? [Login!](#)".



[voteonline](#)[elevators](#)[overview](#)

component

X

+

[← → Q cc/component-us1](#)

User story 1: Login

The screenshot illustrates a user story for a login feature, showing the following components:

- Top Left:** A browser window showing the "WSU VoteOnline" login page. The URL is 127.0.0.1:8000/verify. The page features the Wright State University logo and a "Login" button.
- Top Right:** An email inbox showing an OTP Verification message from wsuvoteonline@gmail.com. The message includes a warning: "This sender wsuvoteonline@gmail.com is from outside your organization." and a link to <https://aka.ms/LearnAboutSenderIdentification>.
- Middle:** A Microsoft Support article titled "Protect yourself from phishing - Microsoft Support". It provides information on identifying phishing scams and includes a link to aka.ms.
- Bottom Left:** A screenshot of a mobile device displaying a verification message: "Check your email for verification".
- Annotations:** A large blue curved arrow points from the "Check your email for verification" message on the mobile device to the Microsoft Support article.

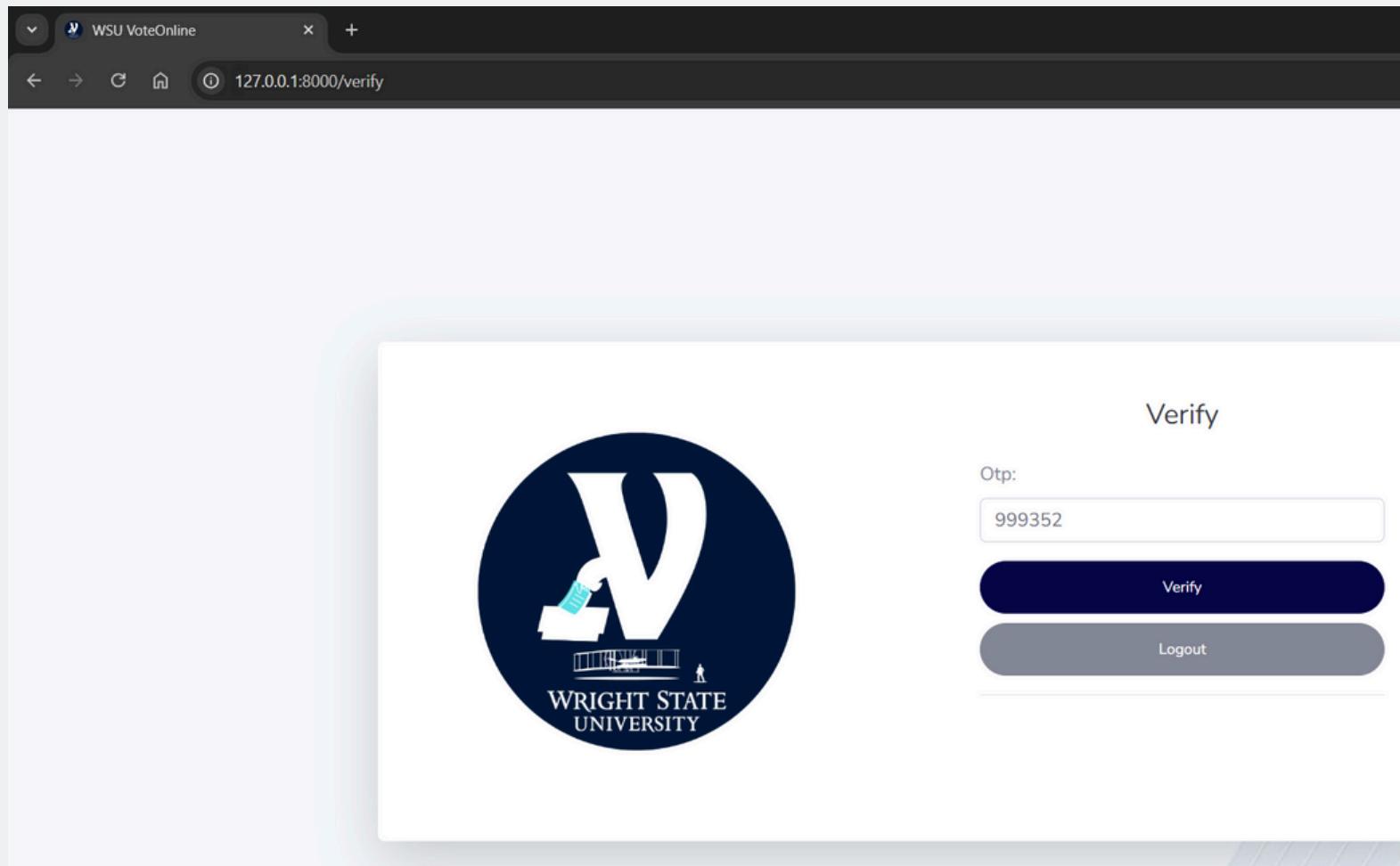
[voteonline](#)[elevators](#)[overview](#)[component](#)

X

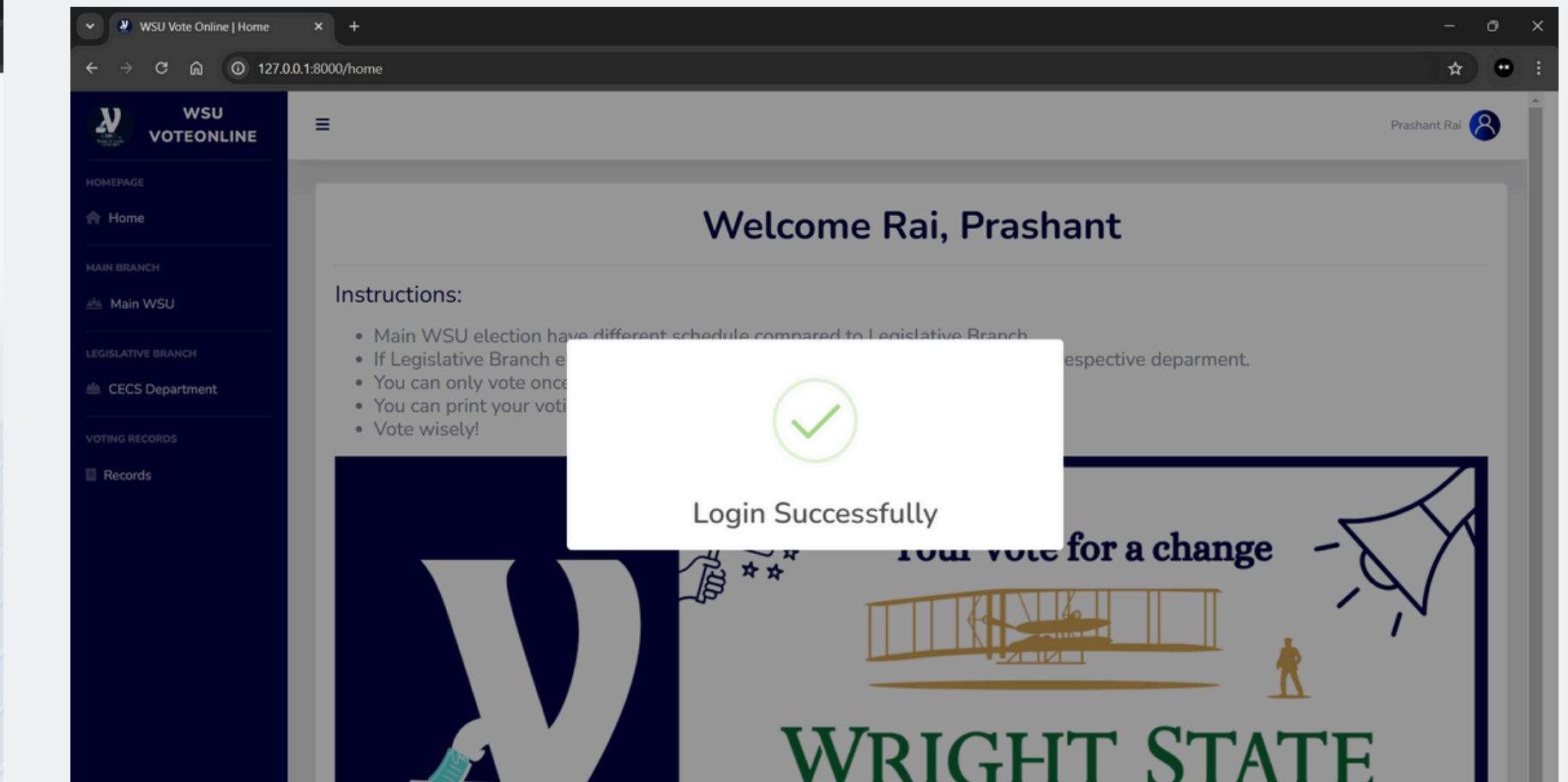
+

[← → Q cc/component-us1](#)

User story 1: Login



A screenshot of a web browser showing the 'Verify' step of the login process. The page features a large Wright State University logo on the left. The main content area has a light gray background with a dark blue header bar containing the word 'Verify'. Below this, there is a text input field labeled 'Otp:' containing the value '999352'. Underneath the input field are two buttons: a dark blue 'Verify' button and a gray 'Logout' button.





voteonline

elevator

Overview

component

X

+



User story 2: As a student, I want to browse through elections and know more about the candidates participating in the elections.

Requirements:

- Navigate to homepage or login.
 - Even guest users can browse through elections and candidates.
 - Registered users can details of candidates.

```
main_candidates = MAIN_Candidate.objects.all()
cecs_candidates = CECS_Candidate.objects.all()
csm_candidates = CSM_Candidate.objects.all()
cla_candidates = CLA_Candidate.objects.all()
cbus_candidates = CBUS_Candidate.objects.all()

all_candidates = list(main_candidates) + list(cecs_cand)

context = {
    'CSM': CSM,
    'CBUS': CBUS,
    'CECS': CECS,
    'CLA': CLA,
    'main': main,
    'today': datetime.date.today(),
    'schedules': schedules,
    'candidates': all_candidates
}
return render(request, 'landingpage.html', context)
```



voteonline

elevator

overview

component



← → Q cc/component-us2

User story 2: Browse Elections

College of Engineering and Computer Science Election

President

James Fry, Cooper Hel, Heman Gosh, Gorgie Fall

Vice President

Hallie Gor, Nicole kulp

Voting Schedule

CECS: Starts: 1d 4h 17m 8s, Ends: 2d 4h 15m 8s

CSM: Starts: 1d 4h 17m 8s

CLA: ELECTION ENDED

User story 2: Browse Candidates

James Fry's Bio

This is James Fry, and I am the most suitable candidate for you. My qualities are better than others.

President

Gorgie Fall, Heman Gosh

Vice President

Hallie Gor, Nicole kulp

CANDIDATES

James Fry, Cooper Hel, Heman Gosh, Hallie Gor, Gorgie Fall, Nicole kulp

ABOUT

This project aims to develop and design a fast, easy and secure online voting system.

← → Q cc/component-us3

User story 3: As an election administrator, I want to create and manage elections, including adding candidates, to ensure the process runs smoothly.

Requirements:

- Login as admin
(for election)
 - Navigate to ‘schedule elections’ and then add elections.
 - Department and start/end date are required.
 - Edit/delete elections
- (for candidate)**
- Navigate to ‘department’ and then add candidate.
 - fill up the candidate details
 - Edit/delete candidates





voteonline

elevator

overview

component

[← → Q cc/component-us3](#)

User story 3: Election management

The screenshot displays two windows of the 'WSU Vote Online | Schedule' application. The left window shows the 'Election Schedule' table with three rows of data:

Department	Election Starts	Election Ends	Action
CECS	Nov. 30, 2024	Dec. 2, 2024	
CSM	Dec. 2, 2024	Dec. 5, 2024	
CLA	Nov. 14, 2024	Nov. 28, 2024	

The right window is a modal titled 'Add Schedule' with fields for 'Department', 'Election Starts', and 'Election Ends'. It includes 'Confirm' and 'Cancel' buttons. A green arrow points from the 'Add' button in the main window to the 'Add Schedule' modal.

Annotations on the interface include:

- A green circle labeled 'Manage election' with a green arrow pointing to the 'Election Schedule' link in the sidebar.
- A blue circle labeled 'Edit Election' with a blue arrow pointing to the edit icon in the table row for the CLA election.
- A red circle labeled 'Delete Election' with a red arrow pointing to the delete icon in the same table row.

voteonline

elevator

overview

component

X

+

← → Q cc/component-us3

User story 3: Candidate management

The screenshot displays two browser windows of the 'WSU Vote Online | CECS Candidates' application. The left window shows a list of candidates for the 'College of Engineering and Computer Science Candidates'. The right window shows an 'Add Candidate' dialog.

Left Window (Candidate List):

- Header:** CECS Department
- Section:** College of Engineering and Computer Science Candidates
- Buttons:** Print, PDF
- Search:** Search: []
- Table Headers:** Position, Photo, Full Name, Bio, Action
- Data:** A list of candidates including President James Fry, President Cooper Hel, President Heman Gosh, President Gorgie Fall, and Vice President Hallie Gor.
- Action Buttons:** For each candidate row, there are two buttons: a blue one with a pencil icon and a red one with a trash bin icon.

Right Window (Add Candidate Dialog):

- Title:** Add Candidate
- Form Fields:**
 - Photo: Choose File [No file chosen]
 - Fullname: [Full name]
 - Bio: Tell us about yourself and your motto
 - Position: [President]
- Buttons:** Confirm, Cancel

Annotations:

- A green arrow points from the 'CECS Candidates' link in the sidebar of the left window to the 'Manage candidate' button at the bottom left.
- A green arrow points from the 'Add' button in the top right of the left window to the 'Edit candidate' button at the bottom center.
- A red arrow points from the 'Delete' button in the bottom right of the left window to the 'Delete candidate' button at the bottom center.
- A green button at the bottom left contains the text 'Manage candidate'.
- A dark blue button at the bottom center contains the text 'Edit candidate'.
- A red button at the bottom center contains the text 'Delete candidate'.

voteonline

elevator

overview

component

X

+

← → Q cc/component-us4

User story 4: As a user, i should be able to post feedbacks

Requirements:

- Navigate to bottom of the homepage
- fill up the form
- confirmation will be sent via email



[voteonline](#)[elevator](#)[overview](#)[component](#)

X

+

[← → Q cc/component-us4](#)

User story 4: Demonstration

The screenshot shows a web page with a header containing the WSU VoteOnline logo, navigation links for Home, About, Services, Elections, and a prominent blue 'VOTE' button. Below this is a section titled 'FeedBack' with the sub-instruction 'Your opion matters to us. Leave us your thoughts.' A text input field contains the email address 'gamer.trill22@gmail.com'. A larger text area contains the feedback message 'VoteOnline is the best!'. At the bottom is a dark blue 'Submit Feedback' button.

Feedback Received Inbox x

[wsuvoteonline@gmail.com](#)

to ▾

Dear User,

Thank you for your feedback. Here is what you submitted:

VoteOnline is the best.

Best regards,
Your Team[Reply](#) [Forward](#) **(User)****(Admin)**

New Feedback Received Inbox x

[wsuvoteonline@gmail.com](#)

to bcc: me ▾

A new feedback has been submitted by [gamer.trill22@gmail.com](#).Feedback:
VoteOnline is the best.

← → Q cc/component-us5

User story 5: As a student, I should be able to vote securely

Requirements:

- Login into the system
- Navigate to department election
- Select desired candidate for each position.
- Confirm vote

```
@login_required(login_url='login')
@verified_or_superuser
@CECS_voter_or_superuser
@department_not_voted_or_superuser
@CECS_schedule_or_superuser
def CECSballot(request):
    context = [
        'title': 'CECS Ballot',
        'president': CECS_Candidate.objects.filter(position='President'),
        'vicepresident': CECS_Candidate.objects.filter(position='Vice President'),
        'secretary': CECS_Candidate.objects.filter(position='Secretary'),
        'treasurer': CECS_Candidate.objects.filter(position='Treasurer'),
        'eventco': CECS_Candidate.objects.filter(position='Event Coordinator'),
        'srofficer': CECS_Candidate.objects.filter(position='SRO Officer'),
        'culturalaffairs': CECS_Candidate.objects.filter(position='Cultural Affairs'),
        'departmenthead': CECS_Candidate.objects.filter(position='Department Head')
    ]
    if request.method == 'POST':
        voter = request.user
        voter.voted_department = True
```

voteonline

elevator

overview

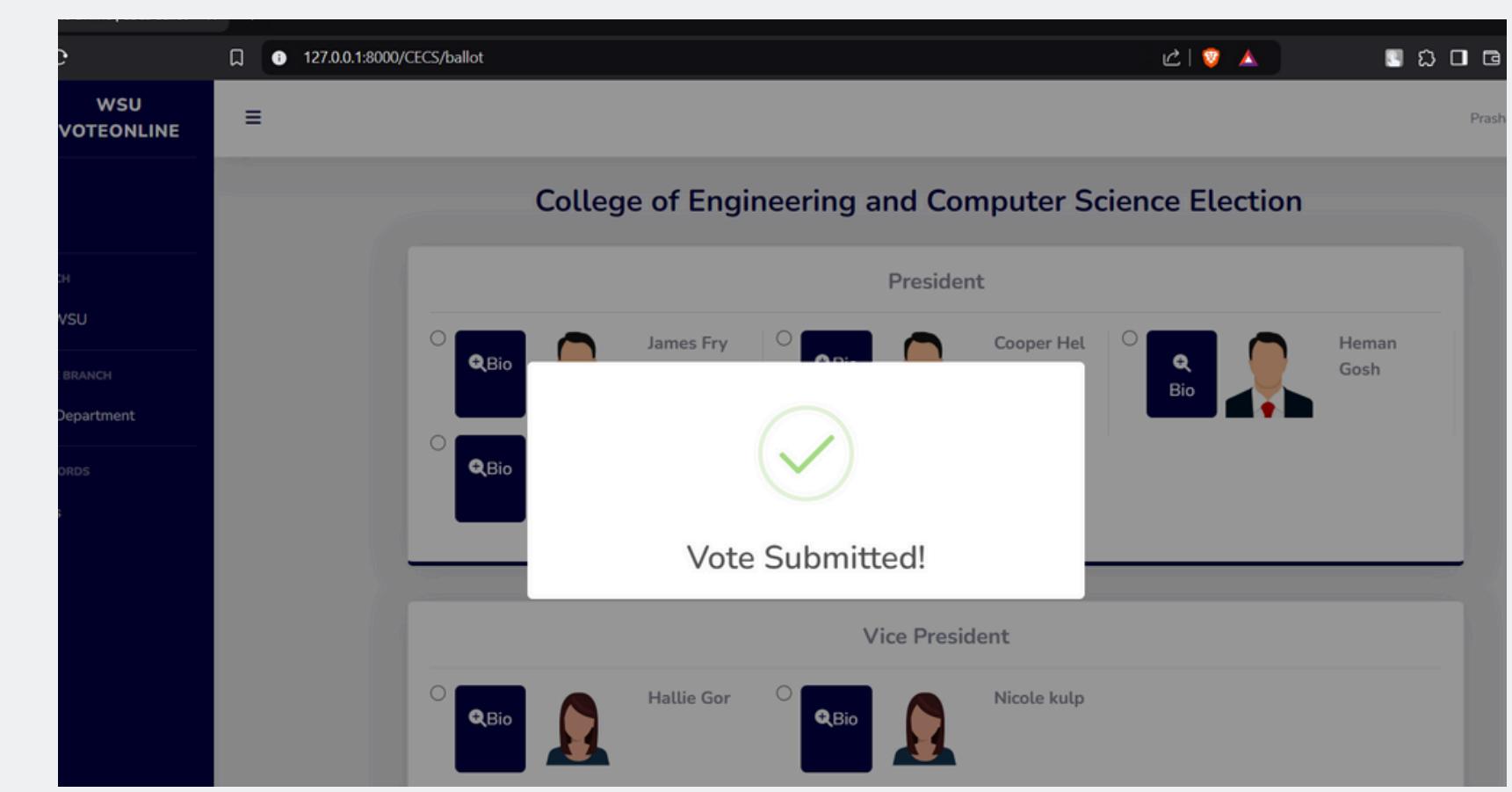
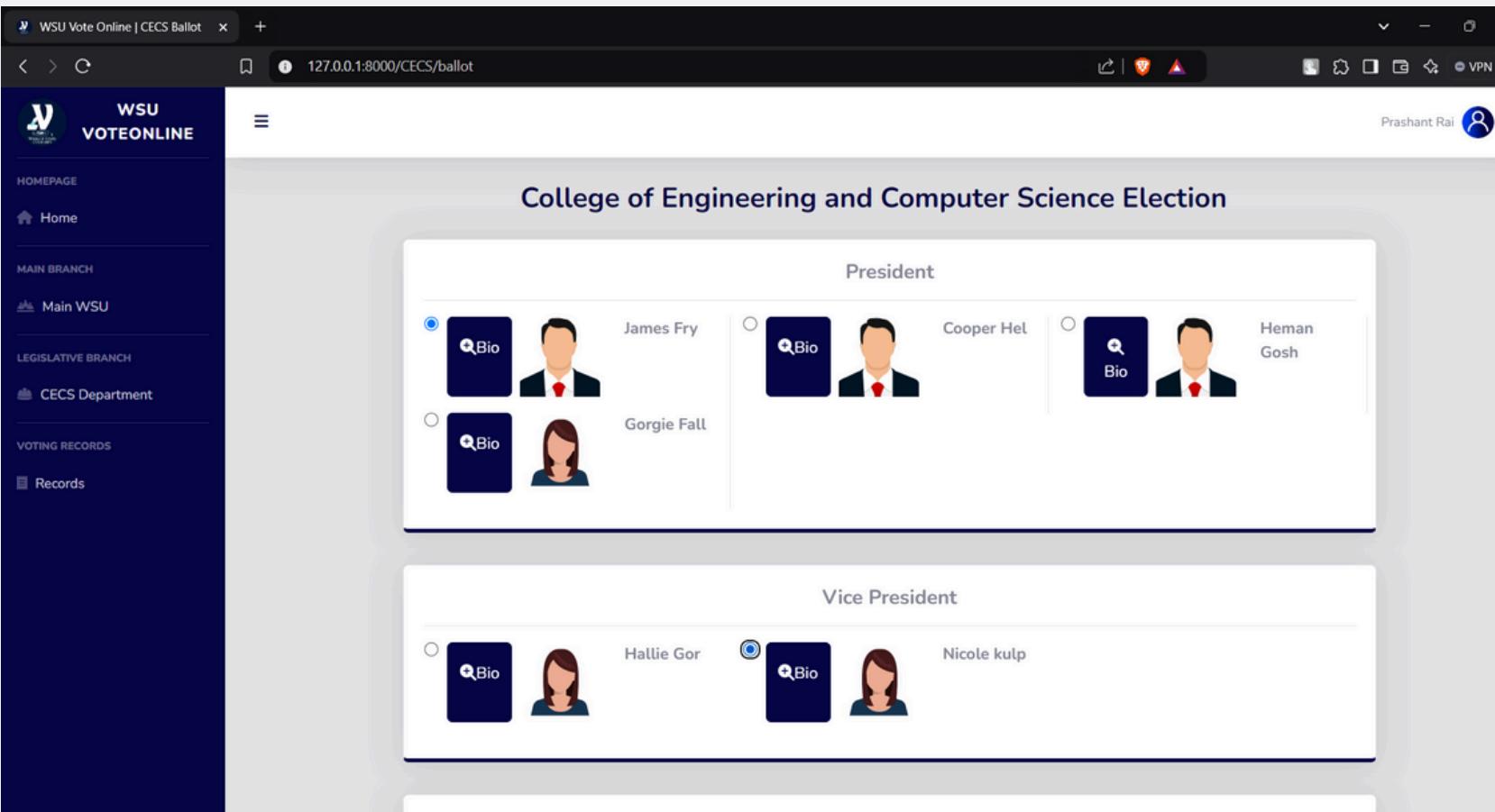
component

X

+

← → Q cc/component-us5

User story 5: Voting ballot



Voting process



Voting success

voteonline

elevator

overview

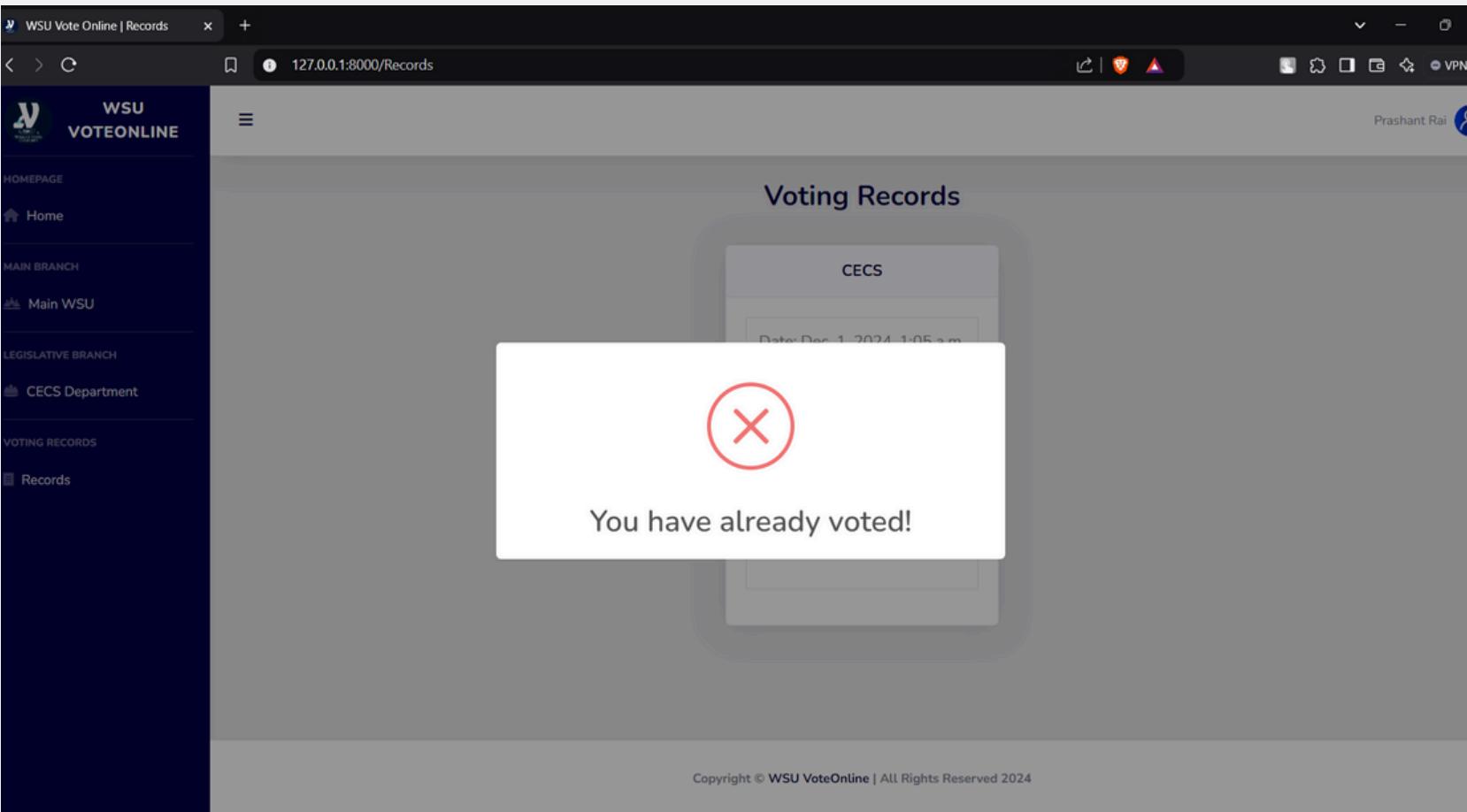
component

X

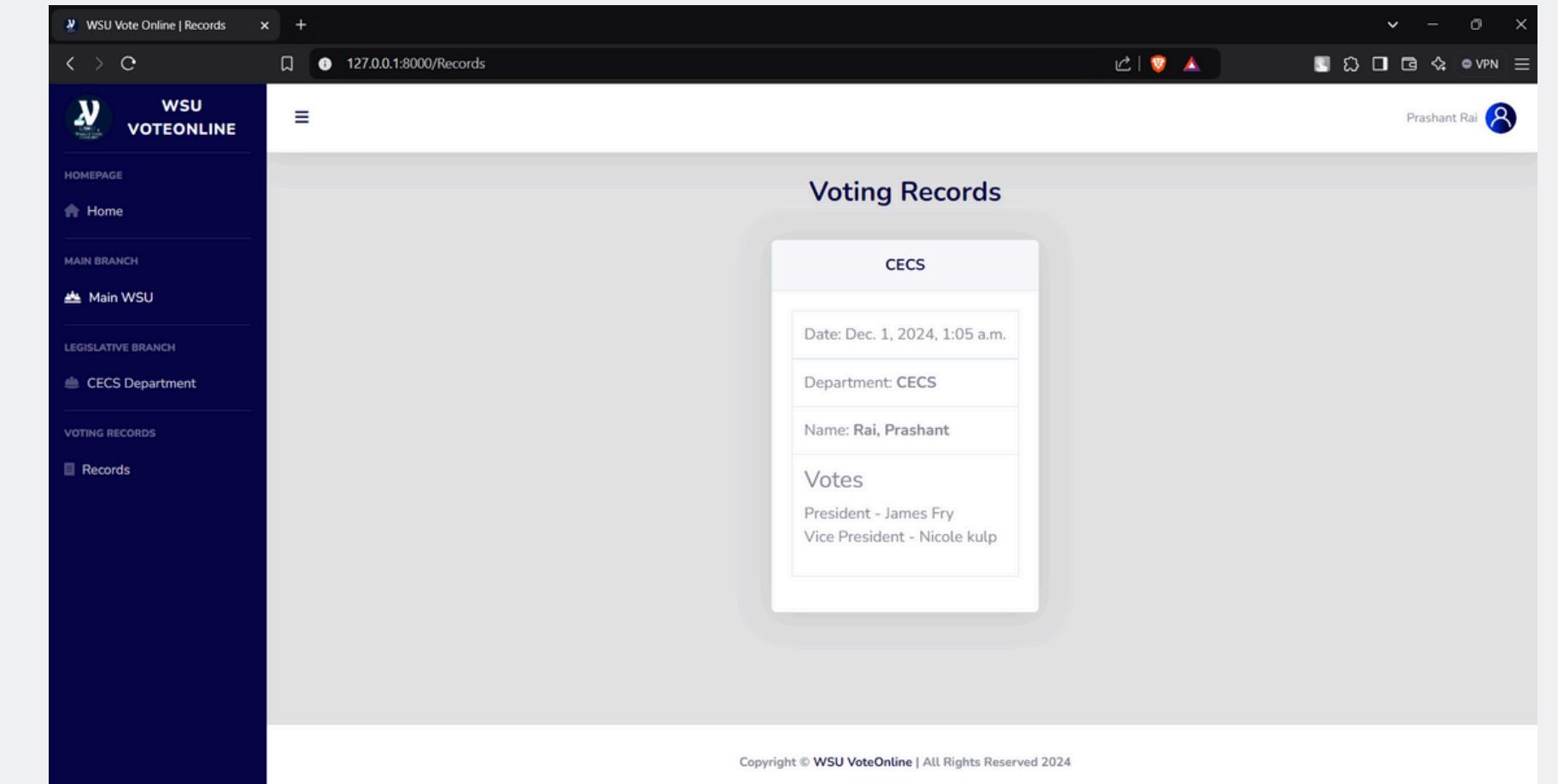
+

← → Q cc/component-us5

User story 5: Voting ballot



Duplicate voting
restrictions



Voting Records

[voteonline](#)[elevator](#)[overview](#)[component](#)[Testing](#)

← → Q cc/testing

Testing

Test Planning

- Unit testing
- Functional testing
- Load testing

Frameworks

- Django unit test framework
- Selenium framework
- Locust framework



[voteonline](#)[elevator](#)[overview](#)[component](#)[testing](#)[conclusion](#)

X +

[← → Q](#) Insert your topic here

Conclusion

Things to improve

- Sometimes, tasks were started later than planned.
- We could have set stricter deadlines for ourselves.

What went well

- The project was actually finished
- Diverse skill set within the group