

Testing

Unit Testing

Test id: 1

Type: Unit Test

Title: Node application running on port 3000

Description: Check if the created node backend application runs successfully and can be opened on port 3000

Input: Node server.js

Expected Outcome: Hello World on localhost:3000

Actual Outcome: Hello World on localhost:3000

Date: 19/01/24

Result: Pass



Hello World

Test id: 2

Type: Unit Test

Title: Synchronize mysql database on port 3006

Description: Synchronize the mysql database with node js server using the sequelize.js framework

Input: Node sequelize.js

Expected Outcome: database synchronized successfully

Actual Outcome: database synchronized successfully (after correcting few syntax errors)

Date: 19/01/24

Result: Pass

```
Executing (default): SHOW INDEX FROM `Elections` FROM `votegrity`
Executing (default): DROP TABLE IF EXISTS `Results`;
Executing (default): CREATE TABLE IF NOT EXISTS `Results` (`id` timestamp NOT NULL, PRIMARY KEY (`id`), FOREIGN KEY (`winner_id`) REFERENCES `Candidates`(`id`));
Executing (default): SHOW INDEX FROM `Results` FROM `votegrity`
Tables synchronized successfully
```

Test id: 3

Type: Unit Test

Title: Register through backend API

Description: Use Postman to register a new user using localhost:3000/api/register

Input:

```
{  
    "name": "user1",  
    "email": "user1@gmail.com",  
    "password": "secretpassword1",  
    "dateOfBirth": "01/01/2001",  
    "specialNumber": "20302222",  
    "citizenship": "Irish",  
    "phoneNumber": "0891234567"  
}
```

Expected Outcome: User Created Successfully

Actual Outcome: User Created Successfully, using
localhost:3000/api/register/api/register, routing url has been fixed successfully
through app.js and userRoute.js

Date: 19/01/24

Result: Pass

```
{  
    "user": {  
        "id": 1,  
        "name": "user1",  
        "email": "user1@gmail.com"  
    },  
    "message": "User created successfully"  
}
```

Test id: 4

Type: Unit Test

Title: Error when registering again

Description: Send an error response back if a user with the same email attempts to
register again

Input:

```
"name": "user1",  
"email": "user1@gmail.com",  
"password": "secretpassword1",  
"dateOfBirth": "01/01/2001",  
"specialNumber": "20302222",  
"citizenship": "Irish",  
"phoneNumber": "0891234567"  
}
```

Expected Outcome: User already exists

Actual Outcome: User already exists

Date: 19/01/24

Result: Pass

```
{  
  "message": "User already exists"  
}
```

Test id: 5

Type: Unit Test

Title: Login through backend API

Description: Use Postman to login a registered user using
localhost:3000/api/user/login

Input:

```
{  
  "email": "user1@gmail.com",  
  "password": "secretpassword1"  
}
```

Expected Outcome: User logged in successfully

Actual Outcome: The login information was incorrect

Date: 20/01/24

Result: Fail

Reason: Each time the server runs, the database is dropped entirely thus having to register a new user each time, not desirable, the login function works but database must be fixed

Actual Outcome: User logged in successfully

Result: Pass

```
{  
  "message": "User logged in successfully"  
}
```

Test id: 6

Type: Unit

Title: Fix Database drop on server startup

Description: The database drops all tables when starting up the server which is undesirable as the data must be kept even when the server is not running

Input: Node server.js

Expected Outcome: Tables are not dropped when server starts

Actual Outcome: Tables are not dropped when server starts

Date: 20/01/24

Result: Pass

Test id: 7

Type: Unit Test

Title: Receive login token from backend API

Description: Use Postman to login a registered user using localhost:3000/api/user/login and receive an authentication token

Input:

```
{  
    "email": "user1@gmail.com",  
    "password": "secretpassword1"  
}
```

Expected Outcome: token

Actual Outcome: token

Date: 20/01/24

Result: Pass

```
        },  
        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ  
            B-o1D4GkETejPUzZf36re3yVt9U5rVKwMjFWVzz-Mjk"  
    }
```

Test id: 8

Type: Unit Test

Title: Authenticate users using tokens on routes

Description: Allow logged in users with tokens to access specific routes, tested using logout function in loginController

Input: send any data to <http://localhost:3000/api/user/logout> on postman

Expected Outcome: message: “logged out”

Actual Outcome: message: “logged out”

Date: 20/01/24

Result: Pass

```
{  
    "message": "Logged out"  
}
```

Test id: 9

Type: Unit Test

Title: Get information about the user

Description: get certain attributes from the Voter table based on the user that is logged in through a get request

Input: n/a

Expected Outcome: user attributes

Actual Outcome:

```
{  
    "email": "user1@gmail.com",  
    "dateOfBirth": "2001-01-01",  
    "specialNumber": "20302222",  
    "citizenship": "Irish",  
    "phoneNumber": "0891234567"  
}
```

Date: 20/01/24

Result: Pass

Test id: 10

Type: Unit Test

Title: Change user email

Description: Send a new email to update and replace with the user's current email

Input:

```
{  
    "email": "user1_1@gmail.com"  
}
```

Expected Outcome: Email updated successfully

Actual Outcome:

```
{  
    "message": "Email updated successfully"  
}
```

Date: 20/01/24

Result: Pass

Test id: 11

Type: Unit Test

Title: Change user phone number

Description: Send a new phone number to update and replace with the user's current number

Input:

```
{  
    "phoneNumber": "0891234568"  
}
```

Expected Outcome: Number updated successfully

Actual Outcome:

```
{  
    "message": "Number updated successfully"  
}
```

Date: 20/01/24

Result: Pass

Test id: 12

Type: Unit Test

Title: Receive 6 digit authentication code to verify registered users

Description: send a get request to receive a 6 digit code the user can use to verify themselves

Input: n/a

Expected Outcome: authToken

Actual Outcome:

```
{  
    "authToken": "544791"  
}
```

Date: 21/01/24

Result: Pass

Test id: 13

Type: Unit Test

Title: Use received authentication token to authenticate new user

Description: send a post request with authToken to authenticate the newly registered user

Input:

```
{  
    "authToken": "331575"  
}
```

Expected Outcome: message: New user successfully authenticated

Actual Outcome:

```
{  
    "message": "New user successfully authenticated"  
}
```

Date: 21/01/24

Result: Pass

Test id: 14

Type: Unit Test

Title: Receive 6 digit authentication code to reset password

Description: send a get request to receive a 6 digit code the user can use change their password

Input: n/a

Expected Outcome: resetToken

Actual Outcome:

```
{
```

```
        "resetToken": "557289"
    }
```

Date: 21/01/24

Result: Pass

Test id: 16

Type: Component Test

Title: Create a new Admin user via terminal

Description: create a new admin by providing an email and password via the terminal command line

Input: email, password

Expected Outcome: Admin created successfully

Actual Outcome: Admin created successfully

Date: 21/01/24

Result: Pass

```
Executing (default): INSERT INTO `Admins` (`id`, `email`, `password`, `cr
Admin created successfully
Shab@HOME:~/2024-ca400-kellyt-46-sidhur-2/src/backend/middleware=>>>
shab@HOME:~/2024-ca400-kellyt-46-sidhur-2/src/backend/middleware=>>>
```

Test id: 17

Type: Unit Test

Title: Login as an admin

Description: login using admin credentials and receive an authentication token

Input: email, password

Expected Outcome: email, token

Actual Outcome: email, token

Date: 22/01/24

Result: Pass

```
{"email": "rishabdev.sidhu2@mail.dcu.ie", "token": "eyJhb
eyJpZCI6MSwiZW1haWwiOiJyaXNoYWJkZXJuY2lkahUyQG1haW
```

Test id: 18

Type: Unit Test

Title: Get list of security questions

Description: get a list of security questions from the SecurityQuestions table

Input: n/a

Expected Outcome: array of questions

Actual Outcome: array of questions

Date: 05/02/24

Result: Pass

```
{  
    "questions": [  
        "What is the name of your pet",  
        "What is your mother's maiden name",  
        "What is your favourite sport",  
        "What is your favourite animal",  
        "Who is your favourite person",  
        "Which city were you born in"  
    ]  
}
```

Test id: 19

Type: Unit Test

Title: Change user password

Description: send a post request to change the user's password

Input:

```
{  
    "securityAnswer1": "oscar",  
    "securityAnswer2": "dog",  
    "resetToken": "571769",  
    "password": "secretPassword1!"  
}
```

Expected Outcome: message: User password updated successfully

Actual Outcome:

```
{  
    "message": "User password updated successfully"  
}
```

Date: 05/03/24

Result: Pass

Test id: 20

Type: Unit Test

Title: Get user status

Description: send a get request to get the current status of the user, whether they are logged in or not, authenticated or an admin

Input: n/a

Expected Outcome: status

Actual Outcome: status

Date: 05/03/24

Result: Pass

```
{  
    "status": "Not Authenticated",  
    "loggedIn": true,  
    "authenticated": true  
}
```

```
{  
    "status": "Not Logged in",  
    "loggedIn": false  
}
```

```
{  
    "status": "Admin",  
    "admin": true,  
    "election": null  
}
```

Test id: 21

Type: Unit Test

Title: Create election

Description: send a post request to create an election with the relevant details

Input:

```
{  
    "title": "Test Election",  
    "description": "this election is being used for testing purposes",  
    "startDate": "2024-02-05T08:00:00",  
    "endDate": "2024-02-20T08:00:00",  
    "resultDate": "2024-02-22T12:00:00",  
    "candidateNumber": "3",  
    "ageRestriction": "15",  
}
```

Expected Outcome: message: User password updated successfully

Actual Outcome:

```
{  
    "election": {  
        "title": "Test Election",  
        "description": "this election is being use",  
        "startDate": "2024-02-05T08:00:00.000Z",  
        "endDate": "2024-02-20T08:00:00.000Z",  
        "resultDate": "2024-02-22T12:00:00.000Z",  
        "candidateNumber": "3",  
        "ageRestriction": "15"  
    },  
    "message": "Election created successfully"  
}
```

Date: 05/03/24

Result: Pass

Test id: 22

Type: Unit Test

Title: Get total number of added candidates

Description: send a get request to see the maximum number of candidates added and how many are currently added

Input: n/a

Expected Outcome: addedCandidates, candidateCount

Actual Outcome:

```
{  
    "addedCandidates": 0,  
    "candidateCount": 3  
}
```

Date: 05/03/24

Result: Pass

Test id: 23

Type: Unit Test

Title: Add a candidate

Description: send a post request to add a candidate with the provided details and an image

Input: form-data

The screenshot shows a form-data input interface. At the top, there are several radio button options: none, form-data (which is selected), x-www-form-urlencoded, raw, binary, and GraphQL. Below this is a table with two columns: 'Key' and 'Value'. The 'Key' column contains 'image', 'name', 'voice', and 'party'. The 'Value' column contains a file selection field with 'trump.jpg' selected, a text field with 'Donald Trump', a text field with 'Make America great again', and a text field with 'republican' respectively. There is also a small preview thumbnail of the uploaded image.

Key	Value
image	File trump.jpg
name	Text Donald Trump
voice	Text Make America great again
party	Text republican

Expected Outcome: Candidate details, message: Candidate created successfully

Actual Outcome:

```
{  
    "candidate": {  
        "isWinner": false,  
        "id": 2,  
        "name": "Donald Trump",  
        "voice": "Make America great again",  
        "party": "republican",  
        "image": {  
            "fieldname": "image",  
            "originalname": "trump.jpg",  
            "encoding": "7bit",  
            "mimetype": "image/jpeg",  
            "destination":  
                "/home/rishab/2024-ca400-kellyt-46-sidhur-2/src/backend/images",  
                "filename": "1707171968878-trump.jpg",  
                "path":  
                    "/home/rishab/2024-ca400-kellyt-46-sidhur-2/src/backend/images/1707171968878-trump.  
jpg",  
                "size": 25669  
        },  
        "dateOfBirth": "1946-06-14",  
        "biography": "blonde",  
        "age": 77  
    }  
}
```

```
        "updatedAt": "2024-02-05T22:26:08.889Z",
        "createdAt": "2024-02-05T22:26:08.889Z"
    },
    "message": "Candidate created successfully"
}
```

Date: 05/03/24

Result: Pass

Test id: 24

Type: Unit Test

Title: Get election results

Description: send a get request to get the information of the election results winner

Input: n/a

Expected Outcome: Candidate winner details

Actual Outcome:

```
{
    "name": "Donald Trump",
    "voice": "Make America great again",
    "party": "republican",
    "image": "[object Object]",
    "biography": "blonde",
    "voteCount": 200
}
```

Test id: 25

Type: Unit Test

Title: Get all candidates

Description: send a get request to get the information of all candidates in the election

Input: n/a

Expected Outcome: array of candidate details

Actual Outcome: array of candidate details

```
{
    "candidates": [
        {
            "id": 2,
            "name": "Donald Trump",
            "voice": "Make America great again",
            "party": "republican"
        }
    ]
}
```

Test id: 26

Type: Unit Test

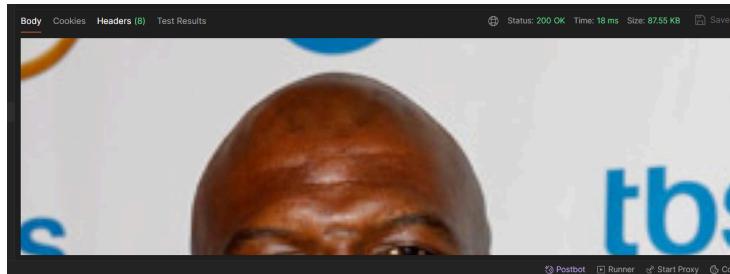
Title: Get candidate image

Description: send a get request to get an image of the candidate with the provided id

Input: n/a

Expected Outcome: Image

Actual Outcome: Image



Test id: 27

Type: Unit

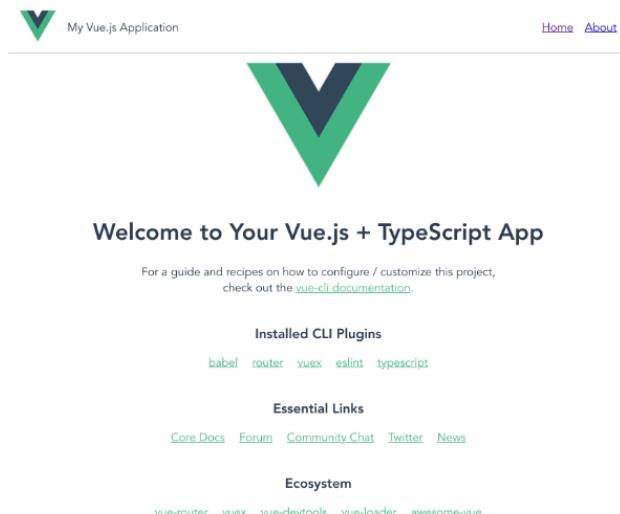
Title: Frontend application installation successful

Description: Ensure the vue js frontend application was successfully installed

Input: npm run serve

Expected Outcome: webpage

Actual Outcome:



Date: 22/01/24

Result: Pass

Test id: 28

Type: Unit

Title: Frontend application router url

Description: Ensure the frontend application routers work successfully when entering the implemented url

Input: n/a

Expected Outcome: webpage

Actual Outcome: webpage

Date: 22/01/24

Result: Pass

Test id: 29

Type: Unit

Title: Title Component

Description: Import the title component and check if it displays the correct route name

Input: n/a

Expected Outcome: Title displayed

Actual Outcome: Title displayed

Date: 25/01/24

Result: Pass

Register

Test id: 30

Type: Unit

Title: Navigation Bar Component

Description: Import the navigation bar component and check it imports each separate button and works correctly

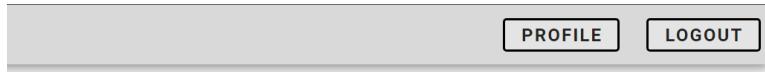
Input: n/a

Expected Outcome: Navigation bar displayed

Actual Outcome: Navigation bar displayed

Date: 25/01/24

Result: Pass



Test id: 32

Type: Unit

Title: Required inputs

Description: Verify that an error is displayed if a required input is empty

Input: empty input

Expected Outcome: error displayed if input is empty

Actual Outcome: error displayed if input is empty

Date: 06/02/24

Result: Pass

Register

Full Name
Full Name is required

Email
Email is required

Password
password is required

Show Password

Citizenship
citizenship is required

Special Number
Special Number is required



Test id: 33

Type: Unit

Title: Code Input

Description: Verify that the code input component renders and stores the entered code

Input: n/a

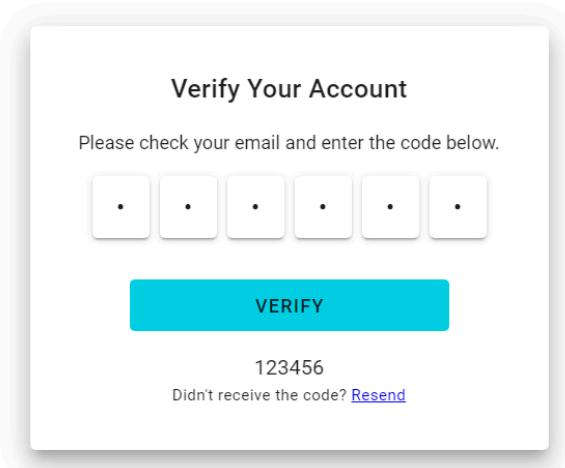
Expected Outcome: component rendered and stores 6 digit code

Actual Outcome: component rendered and stores 6 digit code

Date: 06/02/24

Result: Pass

Authentication



Test id: 34

Type: Unit

Title: Image input

Description: add an image to a form and store the image

Input: image.jpg

Expected Outcome: Image stored in file input

Actual Outcome: Image stored in file input

Date: 06/02/24

Result: Pass



Test id: 35

Type: Unit

Title: Date Input

Description: Use the date picker to add a date

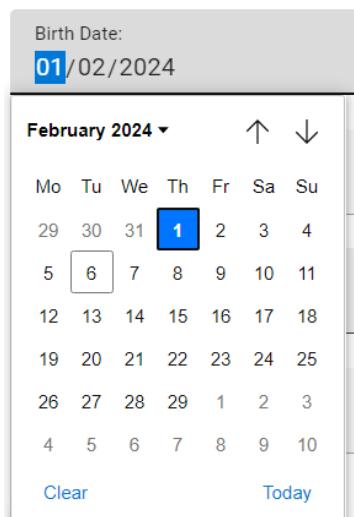
Input: n/a

Expected Outcome: Date chosen and stored

Actual Outcome: Date chosen and stored

Date: 06/02/24

Result: Pass



Test id: 50

Type: Unit

Title: Redirect user from wrong page

Description: If a user visits a page they should not such as the register page when they are logged in

Input: n/a

Expected Outcome: redirected outside of register page

Actual Outcome: redirected outside of register page

Date: 07/02/24

Result: Pass

Test id: 61

Type: Unit

Title: Encrypt password in frontend

Description: Encrypt the original password and send to backend

Input: normal password

Expected Outcome: encrypted password

Actual Outcome: encrypted password

Date: 17/02/24

Result: Pass

Test id: 62

Type: Unit

Title: Decrypt password in backend

Description: Decrypt the received password from the frontend into the correct original password

Input: Encrypted password

Expected Outcome: decrypted password

Actual Outcome: decrypted password

Date: 17/02/24

Result: Pass

Test id: 63

Type: Unit

Title: Success Card Dialog

Description: Verify that the success card dialog box renders correctly and opens a dialog box when prompted to

Input: button submission

Expected Outcome: dialog box opened

Actual Outcome: dialog box opened

Date: 22/02/24

Result: Pass

Test id: 64

Type: Unit

Title: Receive 6 digit delete code via API

Description: receive a 6 digit code from /api/user/deletecode

Input: n/a

Expected Outcome: 6 digit code

Actual Outcome: 6 digit code

Date: 25/02/2024

Result: Pass

Test id: 65

Type: Unit

Title: Delete Account via API

Description: Delete user's account using the 6 digit code received via email by post to /api/user/deleteaccount

Input: 6 digit code

Expected Outcome: user account deleted

Actual Outcome: user account deleted

Date: 25/02/2024

Result: Pass

Test id: 66

Type: Unit

Title: Delete account via form

Description: Fill in the code form with the 6 digit code received via email and submit, redirect to /login where credentials do not work anymore

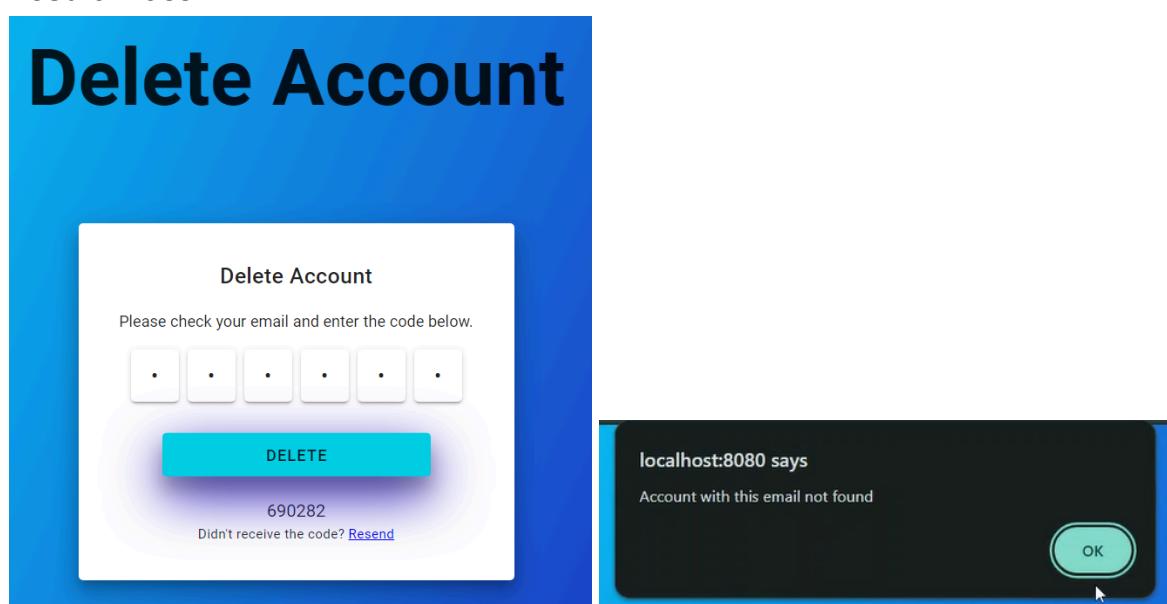
Input: 6 digit code

Expected Outcome: Login credentials don't work after deletion

Actual Outcome: Login credentials don't work after deletion

Date: 25/02/2024

Result: Pass



Test id: 67

Type: Unit

Title: Login rate limiting

Description: throw an error message and disable login for a certain amount of time if too many incorrect attempts are provided in a certain time frame

Input: email, incorrect password

Expected Outcome: error: "Too many login attempts, please try again later."

Actual Outcome:

```
{  
    "error": "Too many login attempts, please try again later.",  
    "time": 1  
}
```

Date: 29/02/2024

Result: Pass

Test id: 68

Type: Unit

Title: Login form rate limiting

Description: throw an error message and disable the login button for a certain amount of time if too many incorrect attempts are provided in a certain time frame

Input: email, incorrect password

Expected Outcome: error: "Too many login attempts, please try again later."

Actual Outcome:

The screenshot shows a blue-themed login interface. At the top, the word 'Login' is displayed in large white font. Below it is a white input field labeled 'Email' containing 'user2@gmail.com'. Underneath is another white input field labeled 'Password' with several dots indicating the password. To the right of the password field is a small checkbox labeled 'Show Password'. Below these fields, a red error message reads 'Too many login attempts, please try again later.' At the bottom of the form are three buttons: 'LOGIN' (disabled), 'RESET' (highlighted in blue), and 'TEST'. Below the buttons are two links: 'Sign Up' and 'Forgot Password?'

Date: 29/02/2024

Result: Pass

Test id: 69

Type: Unit

Title: Assymetric password encryption

Description: Encrypt the user's password in frontend using their public key then decrypt the password in the backend using the user's private key

Input: password

Expected Outcome: logged in

Actual Outcome: logged in

Date: 09/03/2024

Result: Pass

Test id: 72

Type: Unit

Title: Publish Election results

Description: Publish the election using a 10 character key provided by email which will be encrypted, sent then decrypted in the backend

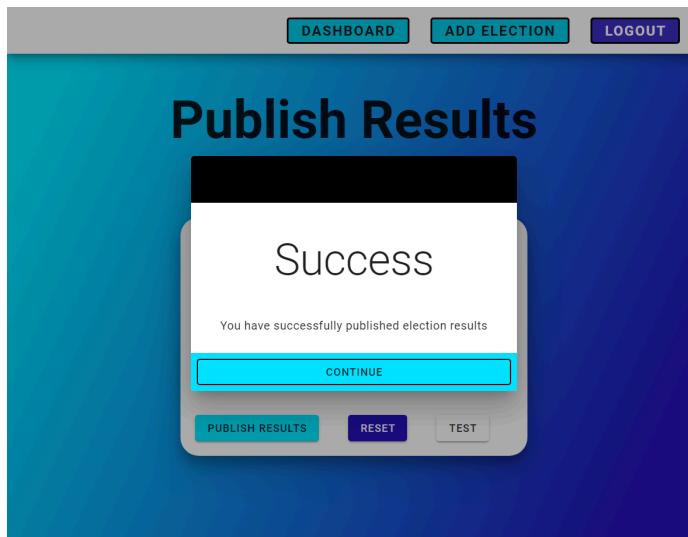
Input: election id, publish key

Expected Outcome: success message

Actual Outcome: success message

Date: 26/03/2024

Result: Pass



Test id: 73

Type: Unit

Title: Deploy Smart Contract

Description: Test if the smart contract is successfully deployed to the blockchain test network

Input: npx hardhat run scripts/deploy.js --network ganache

Expected Outcome: Contract deployed to address: {address}

Actual Outcome: Contract deployed to address: {address}

Date: 15/04/2024

Result: Pass

```
Shab@HOME:~/2024-ca400-kellyt-46-sidhur-2/src/blockchain/contract->>> npx hardhat run scripts/deploy.js --network ganache
WARNING: You are currently using Node.js v19.9.0, which is not supported by Hardhat. This can lead to unexpected behavior. See https://hardhat.org/nodejs-versions

Contract deployed to address: 0x49FCFfaa4b1CF483EB81727033C5636B97d474e7
```

-- BACK	BLOCK 1					
GAS USED 715309	GAS LIMIT 6721975	MINED ON 2024-04-15 22:31:57	BLOCK HASH 0xe4563810934cebd0f71b5abb6f25c551695b1eca59fe9d82a86418da9885d3b			
<hr/>						
TX HASH 0xb89ad025135df2109b138d637d2d65b30f60c496f8d237573fbc1cb081b61638						CONTRACT CREATION

Test id: 74

Type: Unit

Title: One vote restriction

Description: Test that a user can only submit one vote per election

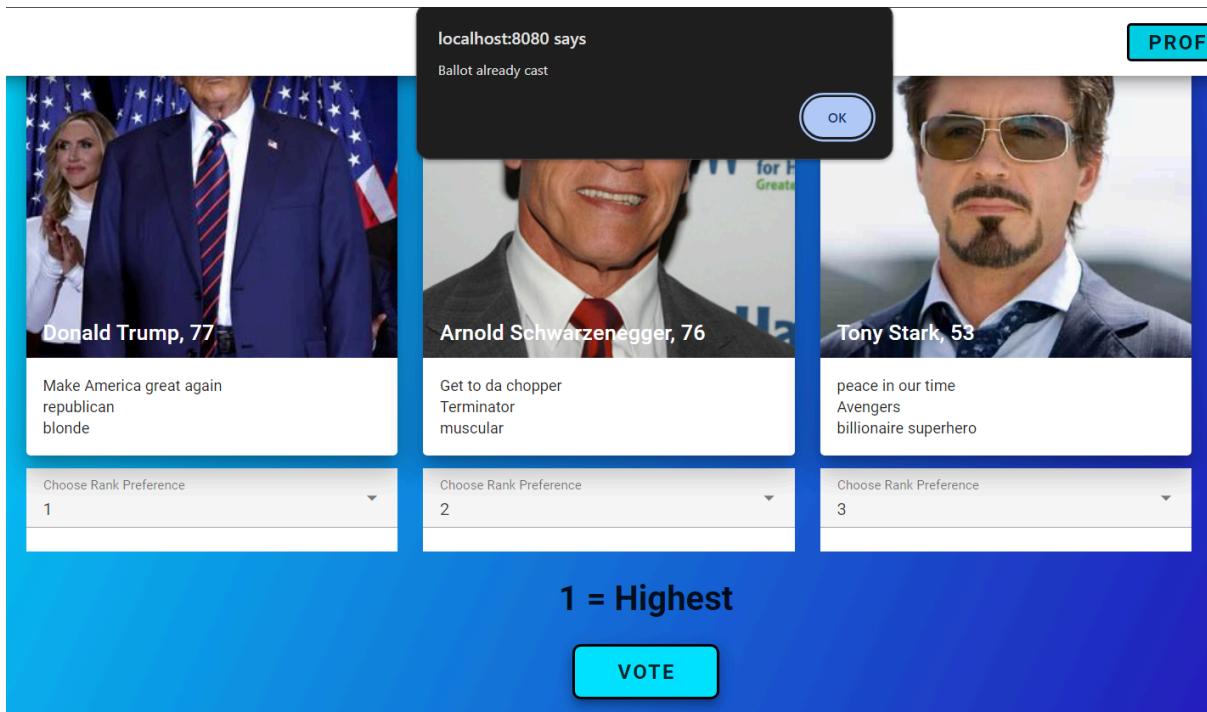
Input: Vote

Expected Outcome: Alert: Ballot already cast

Actual Outcome: Alert: Ballot already cast

Date: 15/04/2024

Result: Pass



Test id: 75

Type: Unit

Title: Encrypt Vote

Description: Check the vote has successfully been combined into one bigint using prime numbers and encrypted

Input: {2: 1, 3: 2, 6: 3}

Expected Outcome: Encrypted BigInt

Actual Outcome: Encrypted BigInt

Date: 15/04/2024

Result: Pass

```
{2: 2, 3: 1, 4: 3}
```

19245411963657564621373304572887498513832560969656658715407959548836441184776584299811896729565522031965525461471770776781603440896046836229871158859222691930953022356579
12021430838257843061054080476452199838849998104921168063350415959452545724599071814842811141381391834641113970165907817210677146486580683459027262659978624348884208717
790155793579964094432719893597223871961111522838109824369504176717535507353865627421056140350756695876619140208894542413421612642673599261857712349370376701058925
3320087561042830851944357513847121295323028785623656719404964401818811029571518109

Test id: 76

Type: Unit

Title: Decrypt Vote

Description: Check the encrypted vote has successfully been decrypted and restored to its original object form with the associated candidateld's and ranks

Input:

Expected Outcome:

Actual Outcome:

Date: 15/04/2024

Result: Pass

192454119636575646213733045
128214298286257842496105496

{2: 2, 3: 1, 4: 3}

Test id: 77

Type: I Init

Type: Unit

Description: Send the encrypted vote to the smart contract using the user's ethereum wallet address.

Input:

Expected Outcome:

Actual Outcome:

Actual Outcome
Date: 15/04/2024

Date: 10/04/22

Ganache

ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES				
CURRENT BLOCK 5	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MERGE	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING	WORKSPACE QUICKSTART	SAVE	SWITCH	⚙️
BLOCK 5	MINED ON 2024-04-12 23:41:13					GAS USED 1481505		1 TRANSACTION		
BLOCK 4	MINED ON 2024-04-12 23:41:13					GAS USED 43472		1 TRANSACTION		
BLOCK 3	MINED ON 2024-04-12 23:40:45					GAS USED 1498665		1 TRANSACTION		
BLOCK 2	MINED ON 2024-04-12 23:40:45					GAS USED 43472		1 TRANSACTION		
BLOCK 1	MINED ON 2024-04-12 23:40:18					GAS USED 681629		1 TRANSACTION		
BLOCK 0	MINED ON 2024-04-12 23:39:20					GAS USED 0		NO TRANSACTIONS		

```
{
  transactionHash: '0x2330f872c412d2af168aa2306f66221e7d546be1d26e10aca4d608f1eb39a134',
  transactionIndex: 0n,
  blockNumber: 3n,
  blockHash: '0x5e1d271963e3cad4620e28fd595501ea4888f73a39b199dfb4fba83919863cde',
  from: '0x33a234185b6fc0ec870a7b172a7250c3ee99cd7f',
  to: '0x49fcffaa4b1cf483eb81727033c5636b97d474e7',
  cumulativeGasUsed: 1498583n,
  gasUsed: 1498583n,
  logs: [
    {
      address: '0x49fcffaa4b1cf483eb81727033c5636b97d474e7',
      blockHash: '0x5e1d271963e3cad4620e28fd595501ea4888f73a39b199dfb4fba83919863cde',
      blockNumber: 3n,
      data: '0x',
      logIndex: 0n,
      removed: false,
      topics: [Array],
      transactionHash: '0x2330f872c412d2af168aa2306f66221e7d546be1d26e10aca4d608f1eb39a134',
      transactionIndex: 0n
    }
  ]
}
```

Test id: 78

Type: Unit

Title: Homomorphic tallying

Description: Tally up the votes using homomorphic multiplication

Input: Array of ranked votes

Expected Outcome: Tallied ranked votes

Actual Outcome: Tallied ranked votes

Date: 15/04/2024

Result: Pass

```
[ { '2': 1, '3': 2, '5': 3 }, { '2': 1, '3': 2, '5': 3 } ]
```

```
{ '2': 2, '5': 4, '6': 6 }
```

Integration Testing

Test id: 15

Type: Integration

Title: Receive 6 digit authentication code via email

Description: send a get request to receive a 6 digit code by email, personal email is used instead of user account email for testing purposes to see if the email is received

Input: n/a

Expected Outcome: email

Actual Outcome: email

Date: 21/01/24

Result: Pass



votegrity.400@gmail.com

to me ▾

Here is your authentication code: 531601

Test id: 31

Type: Integration

Title: Form and Input Component

Description: Import the register form component and check it imports each separate input box and stores the values

Input: n/a

Expected Outcome: input values displayed onto the console

Actual Outcome: input values displayed onto the console

Date: 27/01/24

Result: Pass

Full Name
Email
Password
Citizenship
Special Number
Phone Number
Birth Date dd/mm/yyyy

testemail@gmail.com
myPassword090!
irish
089123456
2006-01-01
oscar
dog

Component Testing

Test id: 36

Type: Component

Title: 404 Page not found

Description: enter an incorrect url path

Input: incorrect url

Expected Outcome: 404 page not found view

Actual Outcome: 404 page not found view

Date: 06/02/24

Result: Pass

404 Page Not Found

Uh oh, Looks like the page you are trying to access does not exist

[RETURN HOME](#)

Test id: 37

Type: Component

Title: Loading screen

Description: Go to home page and check that the loading screen works

Input: localhost:8080/

Expected Outcome: Loading Screen view

Actual Outcome: Loading Screen view

Date: 06/02/24

Result: Pass

Loading...



Test id: 38

Type: Component

Title: Register Page

Description: Verify the Register page and all its imported components render correctly without errors

Input: localhost:8080/register

Expected Outcome: Register Page renders correctly

Actual Outcome: Register Page renders correctly

Date: 06/02/24

Result: Pass

Register

The screenshot shows a registration form with the following fields:

- Full Name
- Email
- Password
- Show Password (checkbox)
- Citizenship
- Special Number

Test id: 39

Type: Component

Title: Login Page

Description: Verify the Login page and all its imported components render correctly without errors

Input: localhost:8080/login

Expected Outcome: Login Page renders correctly

Actual Outcome: Login Page renders correctly

Date: 06/02/24

Result: Pass

Login

The screenshot shows a login form with the following fields:

- Email
- Password
- Show Password (checkbox)
- LOGIN (button)
- RESET (button)
- TEST (button)

Test id: 40

Type: Component

Title: Authentication Page

Description: Verify the Authentication page and all its imported components render correctly without errors

Input: localhost:8080/authentication

Expected Outcome: Authentication Page renders correctly

Actual Outcome: Authentication Page renders correctly

Date: 06/02/24

Result: Pass

The screenshot shows a web page titled "Authentication". At the top right is a "LOGOUT" button. Below the title is a modal window titled "Verify Your Account" with the instruction "Please check your email and enter the code below." It contains a 4x4 grid of input fields for entering a verification code. Below the grid is a blue "VERIFY" button. At the bottom of the modal is a link "Didn't receive the code? [Resend](#)".

Test id: 41

Type: Component

Title: Profile Page

Description: Verify the Profile page and all its imported components render correctly without errors

Input: localhost:8080/profile

Expected Outcome: Profile Page renders correctly

Actual Outcome: Profile Page renders correctly

Date: 06/02/24

Result: Pass

The screenshot shows a web page titled "Profile". At the top right are "VOTE" and "LOGOUT" buttons. Below the title is a form with six input fields: "Name", "Date of Birth", "Special Number", "Citizenship", "Email", and "Phone Number". Each input field has a placeholder text above it.

Test id: 42

Type: Component

Title: Vote Page

Description: Verify the Vote page and all its imported components render correctly without errors

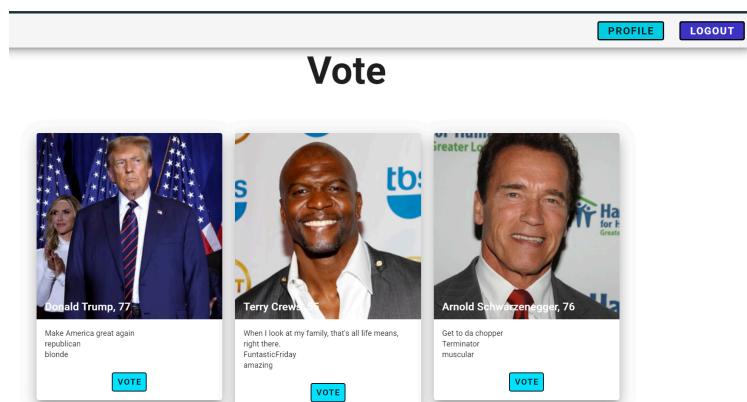
Input: localhost:8080/vote

Expected Outcome: Vote Page renders correctly

Actual Outcome: Vote Page renders correctly

Date: 06/02/24

Result: Pass



Test id: 43

Type: Component

Title: Change Password Page

Description: Verify the Change Password page and all its imported components render correctly without errors

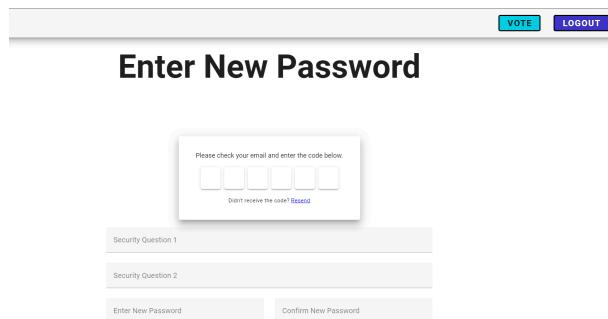
Input: localhost:8080/changepassword

Expected Outcome: Change Password Page renders correctly

Actual Outcome: Change Password Page renders correctly

Date: 06/02/24

Result: Pass



Test id: 44

Type: Component

Title: Thank You Page

Description: Verify the Thank You page and all its imported components render correctly without errors

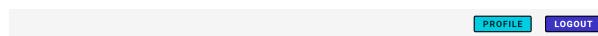
Input: localhost:8080/thankyou

Expected Outcome: Thank You Page renders correctly

Actual Outcome: Thank You Page renders correctly

Date: 06/02/24

Result: Pass



Thank You For Voting

Each vote makes a difference

Test id: 45

Type: Component

Title: Results Page

Description: Verify the Results page and all its imported components render correctly without errors

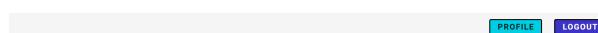
Input: localhost:8080/results

Expected Outcome: Results Page renders correctly

Actual Outcome: Results Page renders correctly

Date: 06/02/24

Result: Pass



Results

Name:	
Voice:	
Party:	
Biography:	



Test id: 46

Type: Component

Title: Admin Login Page

Description: Verify the Admin Login page and all its imported components render correctly without errors

Input: localhost:8080/adminlogin

Expected Outcome: Admin Login Page renders correctly

Actual Outcome: Admin Login Page renders correctly

Date: 06/02/24

Result: Pass

Admin Login

Email

Password

Show Password

LOGIN **RESET** **TEST**

Test id: 47

Type: Component

Title: Dashboard Page

Description: Verify the Dashboard page and all its imported components render correctly without errors

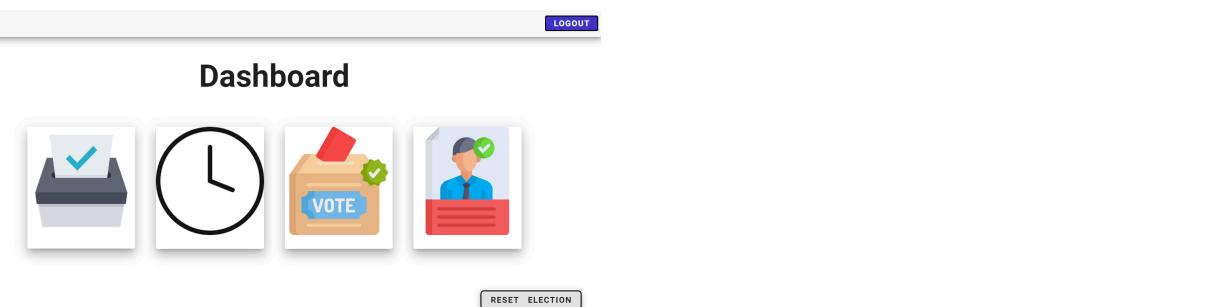
Input: localhost:8080/admin/dashboard

Expected Outcome: Dashboard Page renders correctly

Actual Outcome: Dashboard Page renders correctly

Date: 06/02/24

Result: Pass



Test id: 48

Type: Component

Title: Create Election Page

Description: Verify the Create Election page and all its imported components render correctly without errors

Input: localhost:8080/admin/createelection

Expected Outcome: Create Election Page renders correctly

Actual Outcome: Create Election Page renders correctly

Date: 06/02/24

Result: Pass

LOGOUT

Create Election

Election Title

Description

Election Start Date:
dd/mm/yyyy

Election End Date:
dd/mm/yyyy

Election Results Date:
dd/mm/yyyy

Test id: 49

Type: Component

Title: Add Candidate Page

Description: Verify the Add Candidate page and all its imported components render correctly without errors

Input: localhost:8080/admin/addcandidate

Expected Outcome: Add Candidate Page renders correctly

Actual Outcome: Add Candidate Page renders correctly

Date: 06/02/24

Result: Pass

LOGOUT

Add Candidate

Number of Candidates: 0/0

Candidate Full Name

Candidate Photo

Birth Date:
dd/mm/yyyy

Biography

Voice

System Testing

Test id: 51

Type: System

Title: Register new user

Description: Register a new user through the /register frontend route

Input:

- Name: User Two

- Email: user2@gmail.com
- Password: myPassword2!
- Citizenship: Irish
- Special Number: 20203030
- Phone Number: 089123456
- Date of birth: 12/12/2001
- Security Question 1: What is your favourite sport
- Security Answer 1: football
- Security Question 2: What is your favourite animal
- Security Answer 2: dog

Expected Outcome: new user created

Actual Outcome: new user created

Date: 08/02/24

Result: Pass

The screenshot shows a registration interface on the left and a database table on the right.

Registration Form (Left):

- Full Name: User Two
- Email: user2@gmail.com
- Password: myPassword2!
- Citizenship: Irish
- Special Number: 20203030
- Phone Number: 089123456
- Birth Date: 12/12/2001
- Security Question 1: What is your favourite animal
- Security Answer 1: dog
- Security Question 2: What is your favourite sport
- Security Answer 2: football

Database Table (Right):

	id	name	email
▶	1	user1	user1@gmail.com
*	2	User Two	user2@gmail.com
*	NULL	NULL	NULL

Test id: 52

Type: System

Title: Login

Description: Login through the /login frontend route and receive an authentication token

Input:

- user2@gmail.com
- myPassword2!

Expected Outcome: token received

Actual Outcome: token received

Date: 08/02/24

Result: Pass

The screenshot shows a login form with the following fields and controls:

- Email: user2@gmail.com
- Password: myPassword2!
- Show Password checkbox (checked)
- LOGIN button (highlighted in blue)
- RESET button
- TEST button
- [Sign Up](#)

Below the form, there is a code snippet in a terminal window:

```
index.js??clonedRule.e=script&lang=js:34
> {email: 'user1@gmail.com', token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZC16MSwi
  .AMzV9.RgenJmXfSrxhsICopO4E3HF6p-iYjT9Lmqrr13L_Y'}
```

Test id: 53

Type: System

Title: Authenticate new user

Description: Receive a six digit code via email and authenticate using the given code in /authentication route

Input: 6 digit code

Expected Outcome: user authenticated and redirected

Actual Outcome: user authenticated and redirected

Date: 08/02/24

Result: Pass

Authentication

Verify Your Account

Please check your email and enter the code below.

• I • • • |

VERIFY

747508
Didn't receive the code? [Resend](#)

	id	name	email	authenticated
▶	1	user1	user1@gmail.com	1
	2	User Two	user2@gmail.com	1

Test id: 54

Type: System

Title: View and change profile details

Description: view the user's information and change the available details in /profile route

Input: 089123457

Expected Outcome: user information displayed and details changed

Actual Outcome: user information displayed and details changed

Date: 08/02/24

Result: Pass

Profile

Name: User Two	Name: User Two
Date of Birth: 2001-12-12	Date of Birth: 2001-12-12
Special Number: 20203030	Special Number: 20203030
Citizenship: Irish	Citizenship: Irish
Current Email: user2@gmail.com	Current Email: user2@gmail.com
Current Phone Number: 089123456	Phone Number: <input type="text" value="089123457"/>
SUBMIT	TEST

Current Phone Number:
089123457

Phone Number

Test id: 55

Type: System

Title: Change password

Description: Go through the process of changing a password, redirect to login and login with new credentials in /changepassword route

Input:

- email: user2@gmail.com
- code: code
- Security question 1: dog
- Security question 2: football
- New password: myNewPass2!

Expected Outcome: password changed

Actual Outcome: password changed

Date: 09/02/24

Result: Pass

Enter New Password

Email

Please check your email and enter the code below.

Didn't receive the code? [Resend](#)

Please check your email and enter the code below.

Didn't receive the code? [Resend](#)

What is your favourite animal

What is your favourite sport

Enter New Password

Confirm New Password

Show Password Show Password

*Untitled -
File Edit For
842218

Test id: 56

Type: System

Title: View Election Results

Description: view the results of an election and its contents in /results route

Input: n/a

Expected Outcome: details of winner displayed

Actual Outcome: details of winner displayed

Date: 08/02/24

Result: Pass

Test id: 58

Type: System

Title: Create election

Description: Create a new election through the /admin/createelection route

Input:

- Title: Election 2
- Description: Second Election
- Start date: 09/02/2024
- End date: 19/02/2024
- Result date: 20/02/2024
- Candidate number: 4
- Age restriction: 14

Expected Outcome: New election is created

Actual Outcome: New election is created

Date: 09/02/24

Result: Pass

Create Election

Election Title Election 2	Election Results Date: 20/02/2024									
Description Second Election	Number of Total Candidates 3									
	Minimum Age Requirement (Optional) 14									
Election Results Date: <input type="text" value="dd/mm/yyyy"/>	Email Domain									
CREATE ELECTION RESET TEST										
<table border="1"><thead><tr><th>ID</th><th>Title</th><th>Description</th></tr></thead><tbody><tr><td>1</td><td>Test Election</td><td>this election is being used for testing purposes</td></tr><tr><td>2</td><td>Election 2</td><td>Second Election</td></tr></tbody></table>		ID	Title	Description	1	Test Election	this election is being used for testing purposes	2	Election 2	Second Election
ID	Title	Description								
1	Test Election	this election is being used for testing purposes								
2	Election 2	Second Election								

Test id: 59

Type: System

Title: Add candidate

Description: Add a new candidate for the election through the /admin/addcandidate route

Input:

Expected Outcome: New candidate is added to the system

Actual Outcome: New candidate is added to the system

Date: 09/02/24

Result: Pass

Number of Candidates: 3/4																									
Candidate Full Name Tony Stark																									
Candidate Photo 																									
Birth Date: 29/05/1970																									
Biography billionaire superhero																									
Voice peace in our time																									
Party Avengers																									
ADD CANDIDATE RESET TEST																									
<table border="1"><thead><tr><th>ID</th><th>Name</th><th>Voice</th><th>Party</th><th>IR</th></tr></thead><tbody><tr><td>2</td><td>Donald Trump</td><td>Make America great again</td><td>republican</td><td>1</td></tr><tr><td>3</td><td>Terry Crews</td><td>When I look at my family, that's all life means, ri...</td><td>FuntasticFriday</td><td>1</td></tr><tr><td>5</td><td>Arnold Schwarzenegger</td><td>Get to da chopper</td><td>Terminator</td><td>1</td></tr><tr><td>6</td><td>Tony Stark</td><td>peace in our time</td><td>Avengers</td><td>1</td></tr></tbody></table>	ID	Name	Voice	Party	IR	2	Donald Trump	Make America great again	republican	1	3	Terry Crews	When I look at my family, that's all life means, ri...	FuntasticFriday	1	5	Arnold Schwarzenegger	Get to da chopper	Terminator	1	6	Tony Stark	peace in our time	Avengers	1
ID	Name	Voice	Party	IR																					
2	Donald Trump	Make America great again	republican	1																					
3	Terry Crews	When I look at my family, that's all life means, ri...	FuntasticFriday	1																					
5	Arnold Schwarzenegger	Get to da chopper	Terminator	1																					
6	Tony Stark	peace in our time	Avengers	1																					

Test id: 60

Type: System

Title: Dashboard

Description: View the contents of the election through the admin dashboard

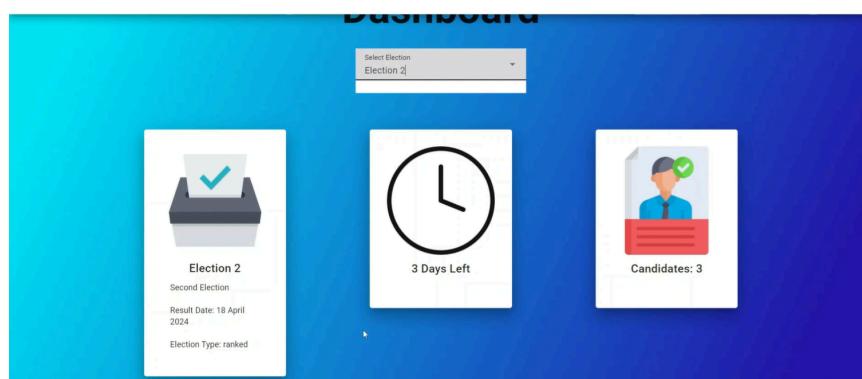
Input: n/a

Expected Outcome: Information about the election is displayed

Actual Outcome: Information about the election is displayed

Date: 09/02/24

Result: Pass



Test id: 79

Type: System

Title:

Description:

Input:

Expected Outcome:

Actual Outcome:

Date: 15/04/2024

Result: Pass

Security Testing

Test id: 70

Type: Security

Title: Flood attacks

Description: Prevent the number of requests

Input: multiple requests in one minute

Expected Outcome: Too many requests from this IP, please try again later.

Actual Outcome:

```
{  
  "error": "Too many requests from this IP, please try again later."  
}
```

}

Date: 20/03/24

Result: Pass

Test id: 71

Type: Security

Title: Zap vulnerability scan

Description: Scan the backend for any possible vulnerabilities

Input: attack localhost:3000

Expected Outcome: empty

Actual Outcome:

▽  Alerts (5)

-  Content Security Policy (CSP) Header Not Set (3)
-  Cross-Domain Misconfiguration (3)
-  Missing Anti-clickjacking Header
-  Server Leaks Information via "X-Powered-By" HTTP Response Header
-  X-Content-Type-Options Header Missing

Date: 21/03/24

Result: Fail

Outcome after error fix:

 Alerts

Result: Pass

User Testing



Assessment

5 responses

[Publish analytics](#)

Name

5 responses

Richard Saini

Patrick Walsh

Katie Doyle

Klaudia Mooney

Aurel Sharku

Age Range

5 responses

 Copy

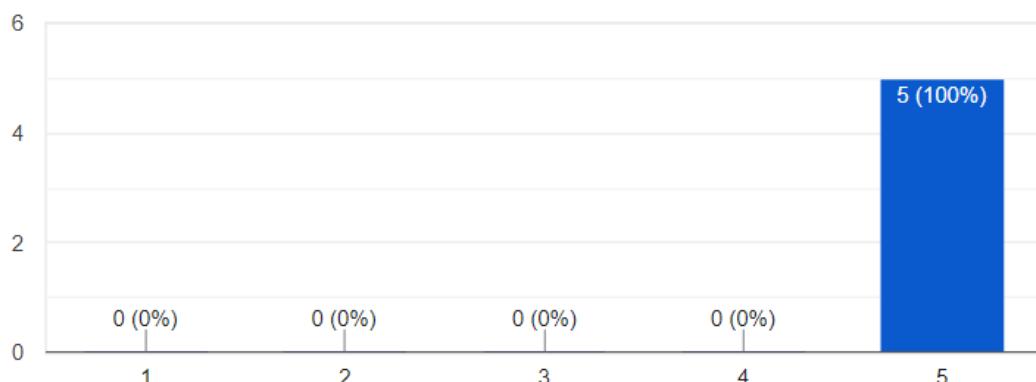


- 18-25
- 26-35
- 36-45
- 46+

What is your level of technical expertise?

5 responses

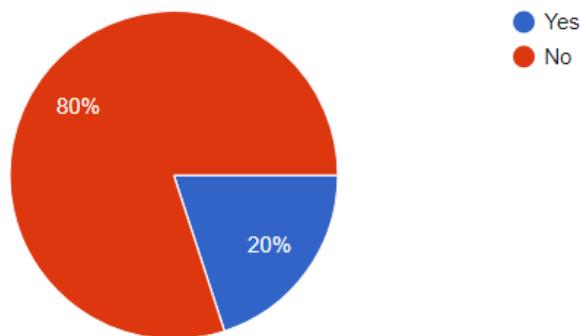
 Copy



Do you know about blockchain technology and how it works?

 Copy

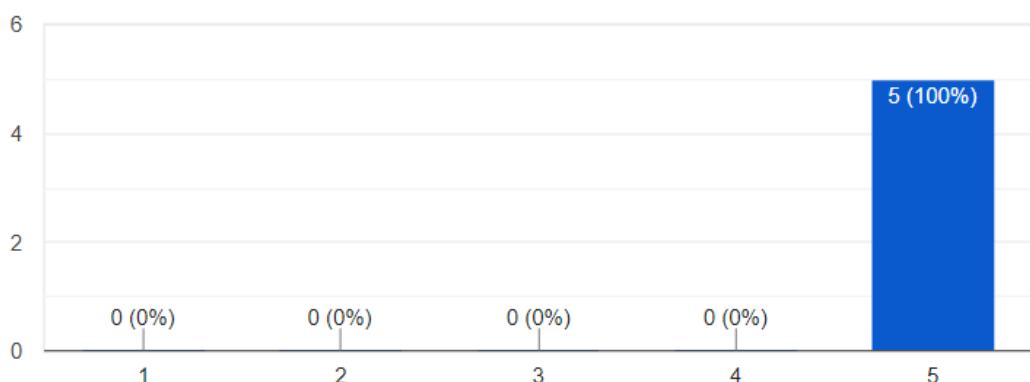
5 responses



How easy was it to navigate through the system

 Copy

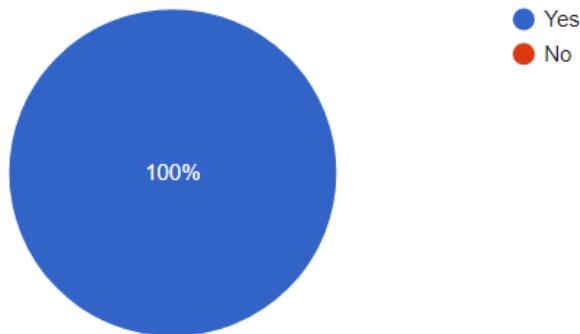
5 responses



Would you be willing to use this system in the real world?

 Copy

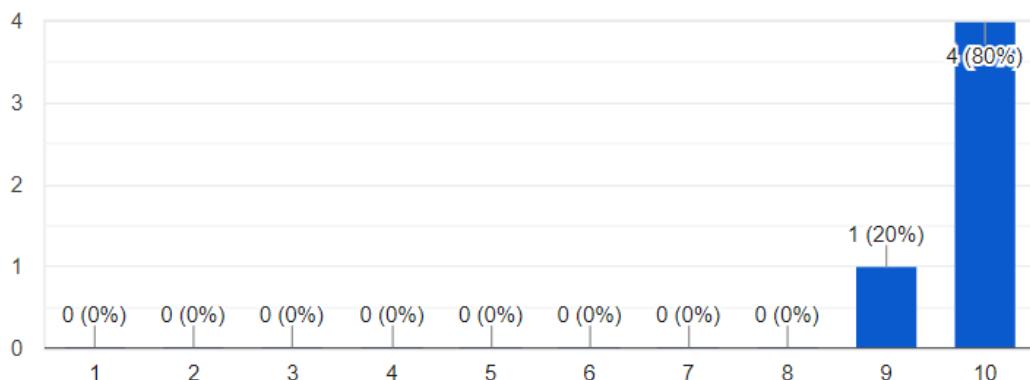
5 responses



On a scale of 1-10 rate your overall experience

 Copy

5 responses



Were there any issues while using the application?

4 responses

Didn't notice anything bad

No

It's a bit hard to see my details on profile, a better way to display that it is not changeable would be nice

no issues

Were there any features that could have been added/improved?

4 responses

maybe a cool animation that shows my vote add to a blockchain

would like to receive codes on sms instead of email

the profile details could be displayed a bit more clearly

everything was good, can't think of any required or useful features to add

Please add any additional comments you may have about the system

3 responses

Very good

Overall very good

Very good and well designed system

I have read the Plain Language Statement (or had it read to me)

 Copy

5 responses

 Yes
 No

100%

I understand the information provided

 Copy

5 responses



 Yes

 No

I have had an opportunity to ask questions and discuss this study

 Copy

5 responses



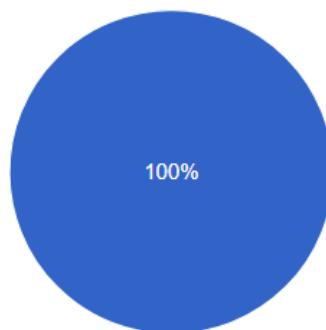
 Yes

 No

I understand the information provided in relation to data protection

 Copy

5 responses



 Yes

 No

I have received satisfactory answers to all my questions

 Copy

5 responses



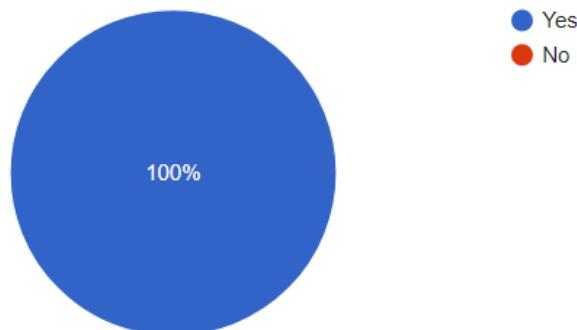
 Yes

 No

I understand I may withdraw from the Research Study at any point

Copy

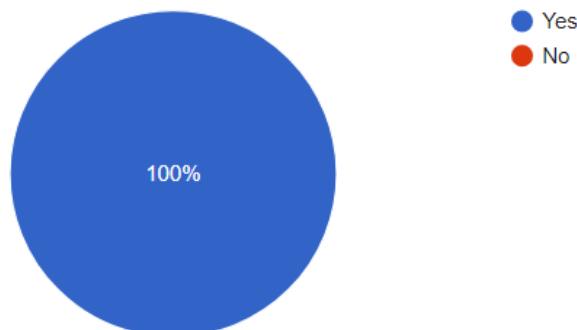
5 responses



I have read and understand the arrangements to be made to protect confidentiality of data, including that confidentiality of information provided is subject to legal limitations

Copy

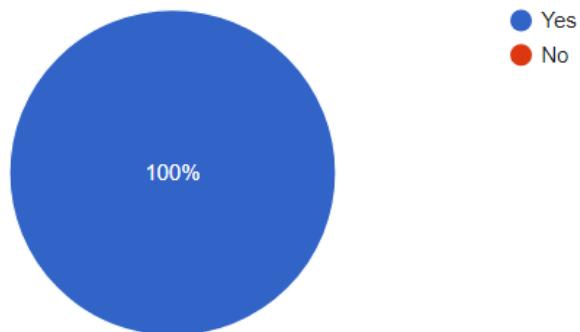
5 responses



I have read and understand confirmations relating to any other relevant information as indicated in the PLS.

Copy

5 responses



I consent to participate in this research study

Copy

5 responses

