

REST API for an Electronic Voting System

Richard Nana Kweenu Quayson

29962024

Department of Computer Science and Information Systems, Ashesi University

CS 341_B: Lab 5 Deploy REST API to Cloud

Dennis Owusu

April 4 2023

Link to repository:

https://github.com/Ashesi-Org/Richard_Quayson_REST_API_LAB_4

Link to Web API:

https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api

1. Register a voter.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/register_voter/`
- Body:** A JSON object:

```
{  "student_id": "29962024",  "firstname": "Timothy",  "lastname": "Quayson",  "email": "richard.quayson@ashesi.edu.gh"}
```
- Response:** 400 Bad Request (1454 ms, 489 B). The response body is:

```
{  "email": "email already exists!",  "student_id": "student_id already exists!"}
```

Figure 1 shows a POST request when an existing `student_id` and email is specified.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/register_voter/`
- Body:** A JSON object:

```
{  "firstname": "Timothy",  "lastname": "Quayson",  "email": "richard.quayson@ashesi.edu.gh"}
```
- Response:** 400 Bad Request (275 ms, 440 B). The response body is:

```
"Student_id is required"
```

Figure 2 shows a POST request when the `student_id` is not specified.

The screenshot shows a REST client interface with the following details:

- Overview:** POST Create Voter Copy
- Environment:** No Environment
- Path:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/register_voter/`
- Method:** POST
- Body:** Empty (raw view)
- Status:** 400 Bad Request, 243 ms, 454 B
- Response Body (JSON):**

```
1 {
2   "message": "Voter information missing!"
3 }
```

Figure 3 shows a POST request when no voter information is provided in the request body.

The screenshot shows a REST client interface with the following details:

- Overview:** POST Create Voter Copy
- Environment:** No Environment
- Path:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/register_voter/`
- Method:** POST
- Body:** Valid JSON (raw view)
- Status:** 201 Created, 537 ms, 540 B
- Response Body (JSON):**

```
1 {
2   "email": "timothy.quayson@ashesi.edu.gh",
3   "firstname": "Timothy",
4   "is_registered": true,
5   "lastname": "Quayson",
6   "student_id": "14712025"
7 }
```

Figure 4 shows a POST request when all required details are present and valid.

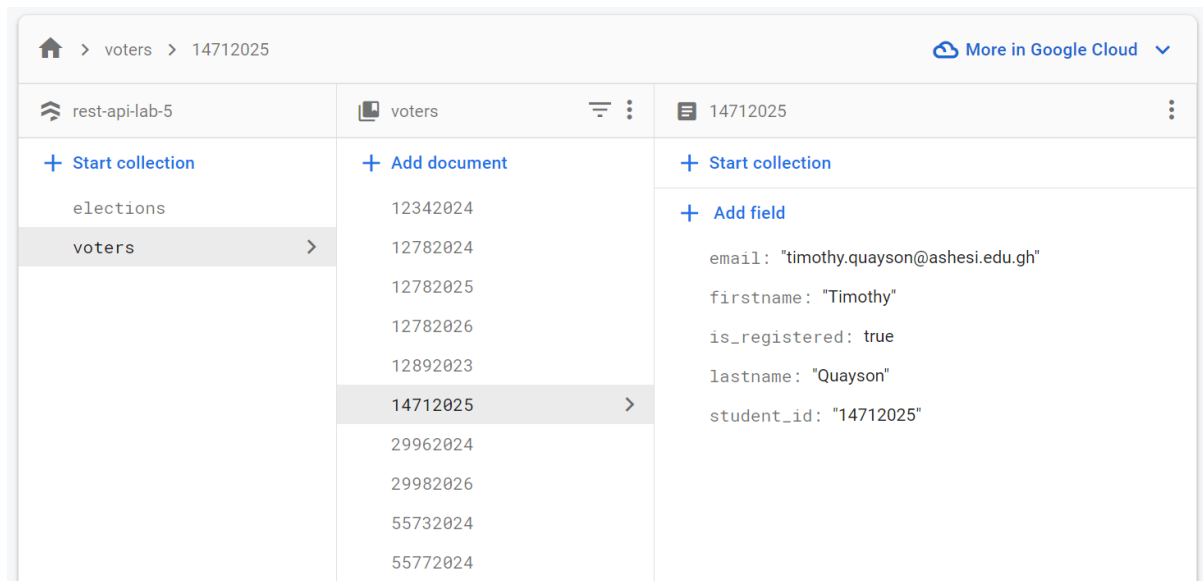


Figure 5 shows the data from the successful POST query above inserted into firebase database.

2. Deregister a voter.

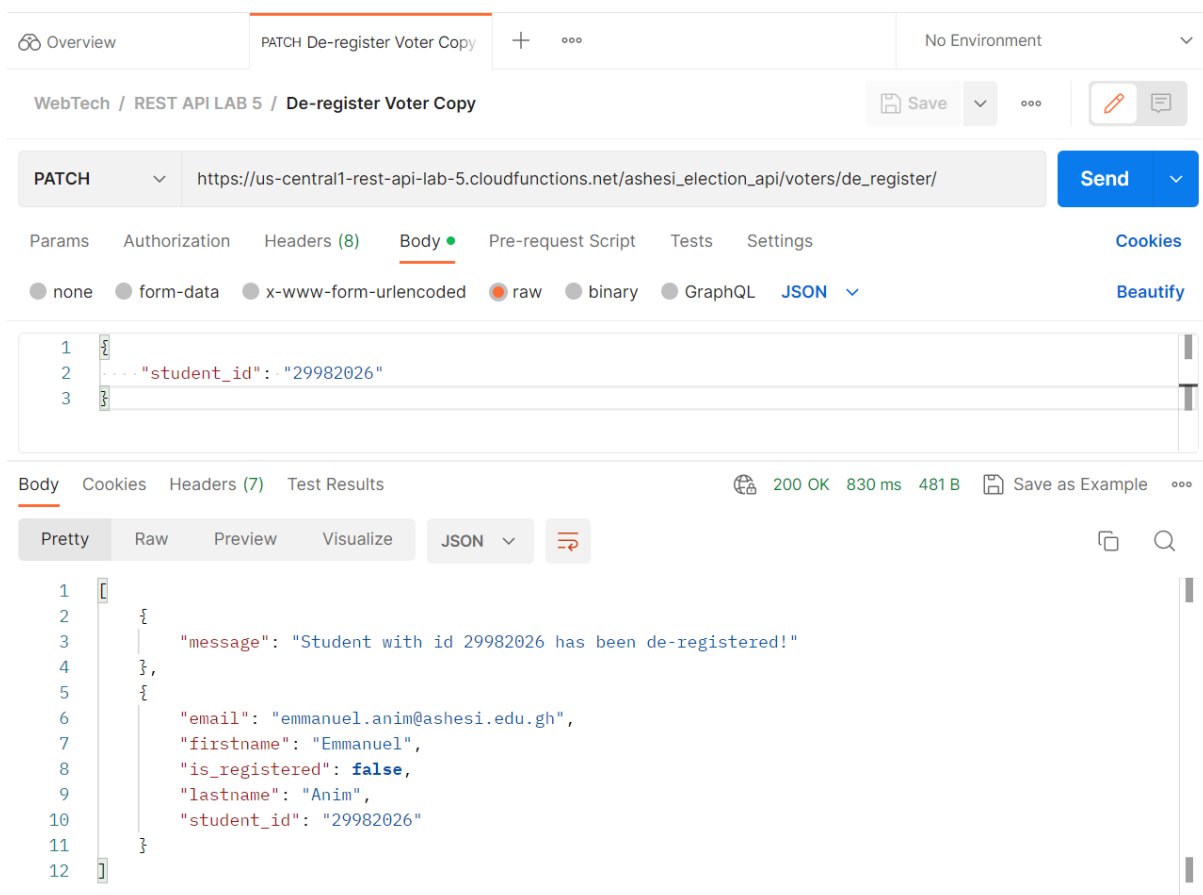


Figure 6 shows a PATCH request to deregister a voter when a valid student id is specified.

The screenshot shows a REST client interface with the following details:

- Method:** PATCH
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/de_register/`
- Body:**

```
{
  "firstname": "Richard"
}
```
- Response:** 400 Bad Request (250 ms, 330 B). The response body is:

```
{
  "message": "Invalid attribute!"
}
```

Figure 7 shows a PATCH request to deregister voters when an invalid attribute is specified.

The screenshot shows a REST client interface with the following details:

- Method:** PATCH
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/de_register/`
- Body:**

```
{
  "year_group": "2025"
}
```
- Response:** 200 OK (509 ms, 615 B). The response body is:

```
{
  "message": "Users in year group 2025 have been de-registered!",
  {
    "email": "elikem.bansah@ashesi.edu.gh",
    "firstname": "Geoffrey",
    "is_registered": false,
    "lastname": "Bansah",
    "student_id": "12782025"
  },
  {
    "email": "timothy.quayson@ashesi.edu.gh",
    "firstname": "Timothy",
    "is_registered": false,
    "lastname": "Quayson",
    "student_id": "14712025"
  }
}
```

Figure 8 shows a PATCH request to deregister voters when a valid year group is specified.

3. Update a voter's details.

The screenshot displays a REST client interface with the following components:

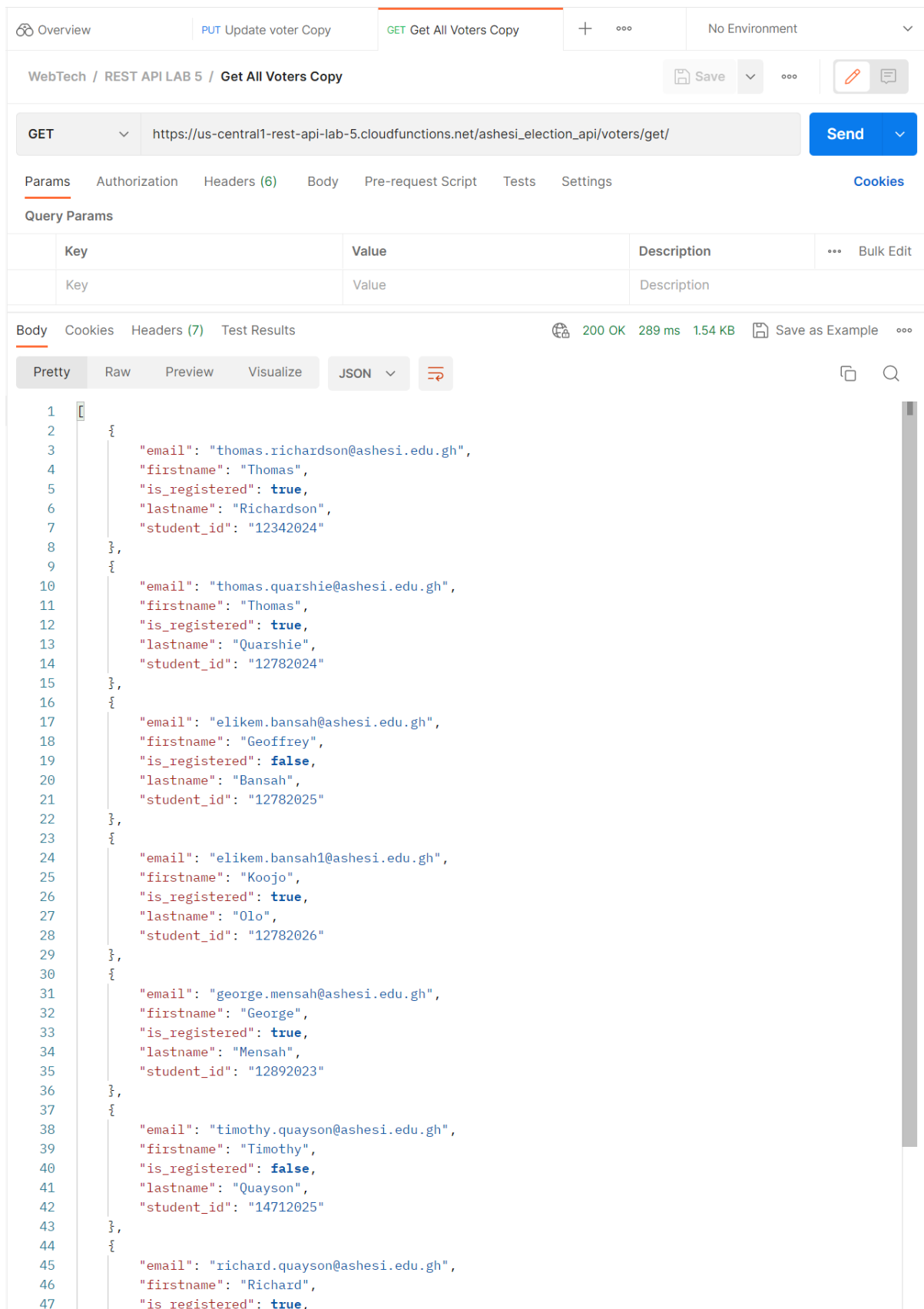
- Overview Tab:** Shows the request method as **PUT** and the URL as `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/update_voter/`. The environment is set to **No Environment**.
- Request Body:** The **Body** tab is selected, showing a JSON payload:

```
1 {
2   "student_id": "14712025",
3   "firstname": "Nana",
4   "is_registered": false,
5   "lastname": "Quayson",
6   "email": "timothy.nquayson@ashesi.edu.gh"
7 }
```
- Response:** The **Body** tab is selected, showing a JSON response:

```
1 {
2   "message": "You cannot use update to deregister, use deregister function instead!"
3 }
```
- Status Bar:** Indicates a **200 OK** status with a response time of **283 ms** and a size of **371 B**. It also includes a **Save as Example** button.

Figure 9 shows a PUT request to update a voter's information when voter is being deregistered.

4. Retrieve a voter's information.



Overview PUT Update voter Copy GET Get All Voters Copy + ... No Environment

WebTech / REST API LAB 5 / Get All Voters Copy Save ...

GET https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

| Key | Value | Description | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body Cookies Headers (7) Test Results 200 OK 289 ms 1.54 KB Save as Example ...

Pretty Raw Preview Visualize JSON

```

1  {
2    {
3      "email": "thomas.richardson@ashesi.edu.gh",
4      "firstname": "Thomas",
5      "is_registered": true,
6      "lastname": "Richardson",
7      "student_id": "12342024"
8    },
9    {
10     "email": "thomas.quarshie@ashesi.edu.gh",
11     "firstname": "Thomas",
12     "is_registered": true,
13     "lastname": "Quarshie",
14     "student_id": "12782024"
15   },
16   {
17     "email": "elitem.bansah@ashesi.edu.gh",
18     "firstname": "Geoffrey",
19     "is_registered": false,
20     "lastname": "Bansah",
21     "student_id": "12782025"
22   },
23   {
24     "email": "elitem.bansah1@ashesi.edu.gh",
25     "firstname": "Koojo",
26     "is_registered": true,
27     "lastname": "Olo",
28     "student_id": "12782026"
29   },
30   {
31     "email": "george.mensah@ashesi.edu.gh",
32     "firstname": "George",
33     "is_registered": true,
34     "lastname": "Mensah",
35     "student_id": "12892023"
36   },
37   {
38     "email": "timothy.quayson@ashesi.edu.gh",
39     "firstname": "Timothy",
40     "is_registered": false,
41     "lastname": "Quayson",
42     "student_id": "14712025"
43   },
44   {
45     "email": "richard.quayson@ashesi.edu.gh",
46     "firstname": "Richard",
47     "is_registered": true,

```

Figure 10 shows a GET request to retrieve all voters.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get All Voters Copy (selected), No Environment.
- WebTech / REST API LAB 5 / Get All Voters Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/?student_id=2996202`
- Params:** student_id: 2996202
- Status:** 400 Bad Request, 811 ms, 336 B
- Body:**

```
{
  "message": "Student ID is not valid."
}
```

Figure 11 shows a GET request to retrieve voter information when an invalid student id is specified.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get All Voters Copy (selected), No Environment.
- WebTech / REST API LAB 5 / Get All Voters Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/?student_id=29962024`
- Params:** student_id: 29962024
- Status:** 200 OK, 314 ms, 421 B
- Body:**

```
{
  "email": "richard.quayson@ashesi.edu.gh",
  "firstname": "Richard",
  "is_registered": true,
  "lastname": "Quayson",
  "student_id": "29962024"
}
```

Figure 12 shows a GET request to retrieve voter information when valid student_id is specified.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get Voters after (selected), GET Get All Voters Copy, +, ... No Environment
- WebTech / REST API LAB 5 / Get Voters after filtering Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/?firstname=Bright`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings, Cookies
- Query Params:**

| Key | Value | Description |
|---|--------|-------------|
| <input checked="" type="checkbox"/> firstname | Bright | |
| Key | Value | Description |

- Body:** Cookies, Headers (7), Test Results
- Response:** 200 OK, 263 ms, 415 B
- Body Content (JSON):**

```
{  "email": "bright.quayson@ashesi.edu.gh",  "firstname": "Bright",  "is_registered": true,  "lastname": "Quayson",  "student_id": "55772024"}
```

Figure 13 shows a GET request to retrieve voter information when their firstname is specified.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get Voters after (selected), GET Get All Voters Copy, +, ... No Environment
- WebTech / REST API LAB 5 / Get Voters after filtering Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/?lastname=qUaY`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings, Cookies
- Query Params:**

| Key | Value | Description |
|--|-------|-------------|
| <input checked="" type="checkbox"/> lastname | qUaY | |
| Key | Value | Description |

- Body:** Cookies, Headers (7), Test Results
- Response:** 200 OK, 807 ms, 680 B
- Body Content (JSON):**

```
[  {    "email": "timothy.quayson@ashesi.edu.gh",    "firstname": "Timothy",    "is_registered": false,    "lastname": "Quayson",    "student_id": "14712025"  },  {    "email": "richard.quayson@ashesi.edu.gh",    "firstname": "Richard",    "is_registered": true,    "lastname": "Quayson",    "student_id": "29962024"  },  {    "email": "bright.quayson@ashesi.edu.gh",    "firstname": "Bright",    "is_registered": true,    "lastname": "Quayson",    "student_id": "55772024"  }]
```

Figure 14 shows a GET request to retrieve voter information when the voter's lastname name begins with quay.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get Voters after (selected), GET Get All Voters Copy, No Environment.
- WebTech / REST API LAB 5 / Get Voters after filtering Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get?email=elikem.bansah@ashesi.edu.gh`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings, Cookies.
- Query Params:**

| Key | Value | Description |
|-------|-----------------------------|-------------|
| email | elikem.bansah@ashesi.edu.gh | |

- Body:** Pretty, Raw, Preview, Visualize, JSON (selected).

```
1 {
2   {
3     "email": "elikem.bansah@ashesi.edu.gh",
4     "firstname": "Geoffrey",
5     "is_registered": false,
6     "lastname": "Bansah",
7     "student_id": "12782025"
8   }
9 }
```
- Status:** 200 OK, 276 ms, 420 B. Save as Example.

Figure 15 shows a GET request to retrieve a voter's information when their email is specified.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get Voters after (selected), GET Get All Voters Copy, No Environment.
- WebTech / REST API LAB 5 / Get Voters after filtering Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get?year_group=2026`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings, Cookies.
- Query Params:**

| Key | Value | Description |
|------------|-------|-------------|
| year_group | 2026 | |

- Body:** Pretty, Raw, Preview, Visualize, JSON (selected).

```
1 [
2   {
3     "email": "elikem.bansah1@ashesi.edu.gh",
4     "firstname": "Koojo",
5     "is_registered": true,
6     "lastname": "Olo",
7     "student_id": "12782026"
8   },
9   {
10    "email": "emmanuel.anim@ashesi.edu.gh",
11    "firstname": "Emmanuel",
12    "is_registered": false,
13    "lastname": "Anim",
14    "student_id": "29982026"
15  }
16 ]
```
- Status:** 200 OK, 257 ms, 541 B. Save as Example.

Figure 16 shows a GET request to retrieve voter information when their year group is specified.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get Voters after (selected), GET Get All Voters Copy.
- WebTech / REST API LAB 5 / Get Voters after filtering Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/?year_group=2026&is_registered=true`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings.
- Query Params:**

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|---------------|-------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | year_group | 2026 | | | |
| <input checked="" type="checkbox"/> | is_registered | true | | | |
| | Key | Value | Description | | |

- Body:** Cookies, Headers (7), Test Results. Status: 200 OK, 323 ms, 414 B. Save as Example.
- Response Body (JSON):**

```
{  "email": "elikeb.bansah1@ashesi.edu.gh",  "firstname": "Koojo",  "is_registered": true,  "lastname": "Olo",  "student_id": "12782026"}
```

Figure 17 shows a GET request to retrieve voter information in the 2026 year group who are registered to vote.

The screenshot shows a REST client interface with the following details:

- Overview:** PUT Update voter Copy, GET Get Voters after (selected), GET Get All Voters Copy.
- WebTech / REST API LAB 5 / Get Voters after filtering Copy**
- Method:** GET
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/voters/get/?lastname=Quayson&is_registered=false&firstname=Richard`
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings.
- Query Params:**

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|---------------|---------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | lastname | Quayson | | | |
| <input checked="" type="checkbox"/> | is_registered | false | | | |
| <input checked="" type="checkbox"/> | firstname | Richard | | | |
| | Key | Value | Description | | |

- Body:** Cookies, Headers (7), Test Results. Status: 404 Not Found, 320 ms, 350 B. Save as Example.
- Response Body (JSON):**

```
{  "message": "No voter found with the provided details"}
```

Figure 18 shows a GET request to retrieve voter information when the details provided does not match any user in the database.

5. Create an election.

The screenshot displays a REST client interface with the following components:

- Overview Tab:** Shows the request method as **POST** and the endpoint as `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/elections/create_election/`. The environment is set to **No Environment**.
- Request Body:** The **Body** tab is selected, showing a JSON payload with the following structure:

```
1 {
2   "election_code": "2023ASC",
3   "election_name": "2023 ASC Election",
4   "election_startdate": "2023-03-27 13:57:30.769268",
5   "election_period": 72,
6   "positions": [
7     {
8       "position_id": "001",
9       "position_name": "ASC President",
10      "candidates": ["12892023", "55732024", "12782025"]
11    },
12    {
13      "position_id": "002",
14      "position_name": "JEC Chairperson",
15      "candidates": ["55772024", "12782026", "29982026"]
16    }
17  ]
18 }
```
- Response:** The **Body** tab is selected, showing the response in **JSON** format:

```
1 {
2   "election_code": "2023ASC",
3   "election_name": "2023 ASC Election",
4   "election_period": 72,
5   "election_startdate": "2023-03-27 13:57:30.769268",
6   "positions": [
7     {
8       "candidates": [
9         {
10          "candidate_id": "12892023",
11          "candidate_voters": []
12        },
13         {
14          "candidate_id": "55732024",
15          "candidate_voters": []
16        },
17         {
18          "candidate_id": "12782025",
19          "candidate_voters": []
20        }
21      ],
22      "position_id": "001",
23      "position_name": "ASC President"
24    },
25    {
26      "candidates": [
27        {
28          "candidate_id": "55772024",
```
- Status:** The response status is **200 OK** with a response time of **334 ms** and a response size of **878 B**.

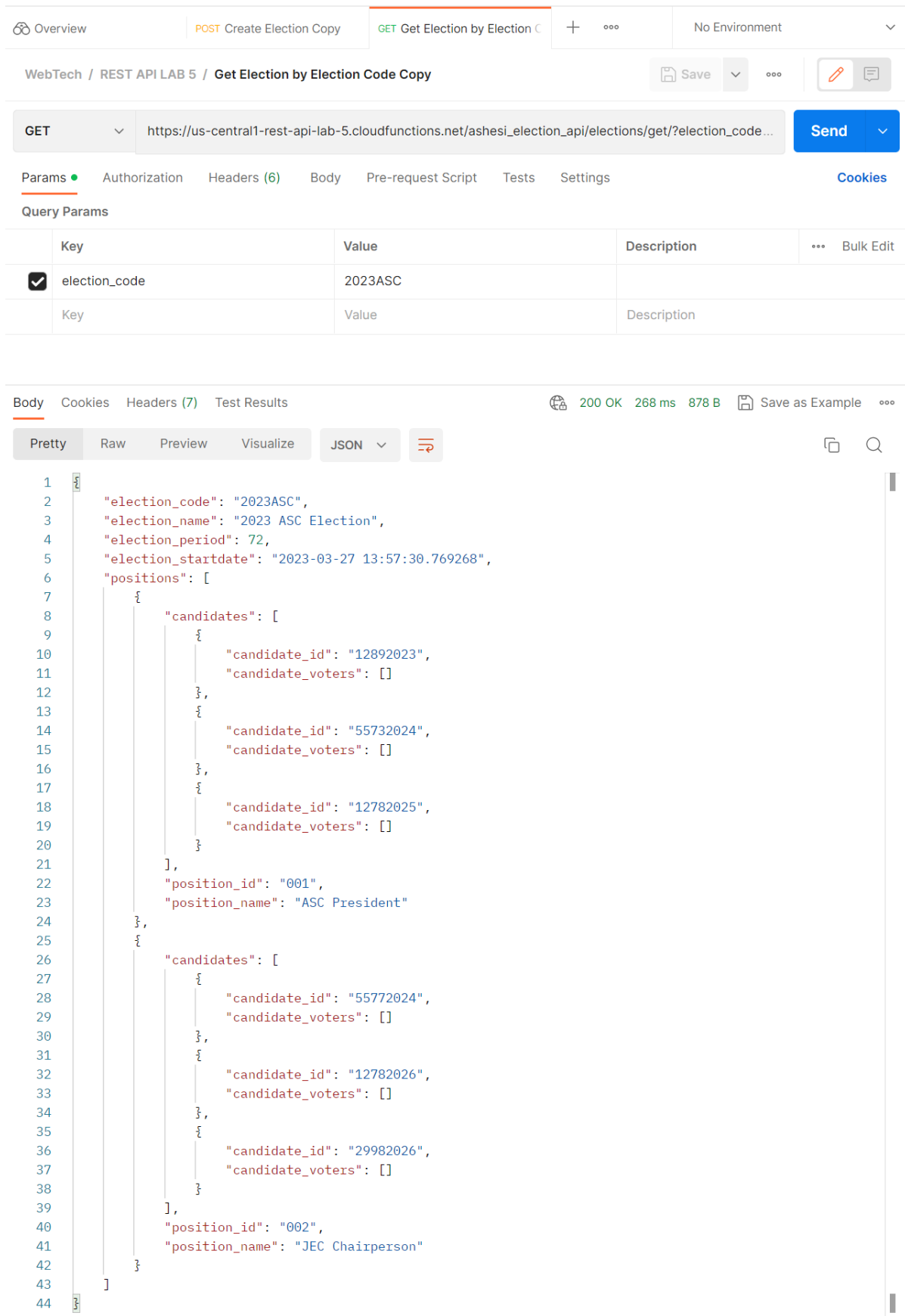
Figure 19 shows a POST request to create an election when all details are valid and unique constraints are met.

The screenshot shows the Firebase console interface. At the top, there are tabs for 'Panel view' and 'Query builder'. Below the tabs, the breadcrumb navigation shows 'home > elections > 2023ASC'. On the right, there is a link 'More in Google Cloud'. The main area is divided into three panels. The left panel shows the 'elections' collection with a 'voters' document. The middle panel shows the '2023ASC' document. The right panel shows the document's fields: 'election_code: "2023ASC"', 'election_name: "2023 ASC Election"', 'election_period: 72', 'election_startdate: "2023-03-27 13:57:30.769268"', and a nested 'positions' array. The 'positions' array has two elements, each with a 'candidates' array and a 'candidate_voters' array. The first position has a candidate with ID '12892023' and three voters with IDs '55732024', '12782025', and '001'. The second position has a candidate with ID '55772024' and two voters with IDs '12782026' and '001'.

| election_code | election_name | election_period | election_startdate | positions |
|---------------|---------------------|-----------------|------------------------------|--|
| "2023ASC" | "2023 ASC Election" | 72 | "2023-03-27 13:57:30.769268" | <ul style="list-style-type: none">0<ul style="list-style-type: none">candidates<ul style="list-style-type: none">0<ul style="list-style-type: none">candidate_id: "12892023"candidate_voters<ul style="list-style-type: none">1<ul style="list-style-type: none">candidate_id: "55732024"candidate_voters<ul style="list-style-type: none">2<ul style="list-style-type: none">candidate_id: "12782025"candidate_voters<ul style="list-style-type: none">position_id: "001"position_name: "ASC President"1<ul style="list-style-type: none">candidates<ul style="list-style-type: none">0<ul style="list-style-type: none">candidate_id: "55772024"candidate_voters<ul style="list-style-type: none">1<ul style="list-style-type: none">candidate_id: "12782026"candidate_voters |

Figure 20 shows the firebase database for the election collection after creating a new election.

6. Retrieve an election.



Overview | POST Create Election Copy | GET Get Election by Election C | + | ... | No Environment

WebTech / REST API LAB 5 / Get Election by Election Code Copy | Save | ... | [Edit] [Comments]

GET | https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/elections/get/?election_code=2023ASC | Send

Params | Authorization | Headers (6) | Body | Pre-request Script | Tests | Settings | Cookies

Query Params

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|---------------|---------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | election_code | 2023ASC | | | |
| | Key | Value | Description | | |

Body | Cookies | Headers (7) | Test Results | 200 OK 268 ms 878 B | Save as Example | ...

Pretty | Raw | Preview | Visualize | JSON | [Copy] [Search]

```

1  {
2    "election_code": "2023ASC",
3    "election_name": "2023 ASC Election",
4    "election_period": 72,
5    "election_startdate": "2023-03-27 13:57:30.769268",
6    "positions": [
7      {
8        "candidates": [
9          {
10             "candidate_id": "12892023",
11             "candidate_voters": []
12           },
13           {
14             "candidate_id": "55732024",
15             "candidate_voters": []
16           },
17           {
18             "candidate_id": "12782025",
19             "candidate_voters": []
20           }
21         ],
22         "position_id": "001",
23         "position_name": "ASC President"
24       },
25       {
26         "candidates": [
27           {
28             "candidate_id": "55772024",
29             "candidate_voters": []
30           },
31           {
32             "candidate_id": "12782026",
33             "candidate_voters": []
34           },
35           {
36             "candidate_id": "29982026",
37             "candidate_voters": []
38           }
39         ],
40         "position_id": "002",
41         "position_name": "JEC Chairperson"
42       }
43     ]
44   }

```

Figure 21 shows a GET request to retrieve election information when the election's code is specified.

7. Delete an election.

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/elections/delete_election/`
- Body:**

```
{
  "election_code": "2023ASC"
}
```
- Response:**

```
{
  "message": "Election with code 2023ASC has been deleted successfully!"
}
```
- Status:** 200 OK, 412 ms, 360 B

Figure 22 shows a DELETE request to delete an election object when the election's code is specified.

The screenshot shows the Firebase database console with the following structure:

- Database:** rest-api-lab-5
- Collection:** elections
- Documents:** The 'elections' collection is currently empty, showing only the collection name and a 'Start collection' button.

Figure 23 shows the firebase database of the election collection after deleting the election.

8. Vote in an election.

The screenshot displays a REST client interface with the following components:

- Overview Tab:** Shows the request method as **POST** and the URL: `https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/elections/vote/2023ASC/?position_id=001`.
- Request Body:** The **Body** tab is selected, showing a JSON payload:

```
1 {
2   "student_id": "55772024",
3   "election_code": "2023ASC",
4   "candidate_id": "12892023"
5 }
```
- Response View:** The **Body** tab is selected, showing the JSON response in **Pretty** format:

```
1 {
2   "election_code": "2023ASC",
3   "election_name": "2023 ASC Election",
4   "election_period": 72,
5   "election_startdate": "2023-03-27 13:57:30.769268",
6   "positions": [
7     {
8       "candidates": [
9         {
10          "candidate_id": "12892023",
11          "candidate_voters": [
12            "55772024"
13          ]
14        },
15        {
16          "candidate_id": "55732024",
17          "candidate_voters": []
18        },
19        {
20          "candidate_id": "12782025",
21          "candidate_voters": []
22        }
23      ],
24      "position_id": "001",
25      "position_name": "ASC President"
26    },
27    {
28      "candidates": [
29        {
30          "candidate_id": "55772024",
31          "candidate_voters": []
32        },
33        {
34          "candidate_id": "12782026",
35          "candidate_voters": []
36        },
37        {
38          "candidate_id": "29982026",
39          "candidate_voters": []
40        }
41      ],
42      "position_id": "002",
43      "position_name": "JEC Chairperson"
44    }
45  ]
46 }
```

Figure 24 shows a POST request for voting in an election when all details are specified.

Overview | **POST Vote in an Election Copy** | + | ... | No Environment

WebTech / REST API LAB 5 / **Vote in an Election Copy** | Save | ... | Edit | Comments

POST | https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/elections/vote/2023ASC/?po... | **Send** | ...

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings | Cookies

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **JSON** | Beautify

```
1 {
2   ... "student_id": "55772024",
3   ... "election_code": "2023ASC",
4   ... "candidate_id": "12892023"
5 }
```

Body | Cookies | Headers (7) | Test Results | 403 Forbidden 877 ms 349 B | Save as Example | ...

Pretty | Raw | Preview | Visualize | **JSON** | ...

```
1 {
2   "message": "You cannot vote twice for one position!"
3 }
```

Figure 25 shows a POST request for voting in an election when the voter has already voted for that position.

Overview | **POST Vote in an Election Copy** | + | ... | No Environment

WebTech / REST API LAB 5 / **Vote in an Election Copy** | Save | ... | Edit | Comments

POST | https://us-central1-rest-api-lab-5.cloudfunctions.net/ashesi_election_api/elections/vote/2023ASC | **Send** | ...

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings | Cookies

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **JSON** | Beautify

```
1 {
2   ... "student_id": "55772024",
3   ... "election_code": "2023ASC",
4   ... "candidate_id": "12892023"
5 }
```

Body | Cookies | Headers (7) | Test Results | 400 Bad Request 2.54 s 338 B | Save as Example | ...

Pretty | Raw | Preview | Visualize | **JSON** | ...

```
1 {
2   "message": "Missing election position!"
3 }
```

Figure 26 shows a POST request to vote in an election when the position is not being specified.