



I have the following html code, which implements with the flask python file also below. I am currently getting a "Method Not Allowed" error:

Graphical Data

Send Message

```
@app.route('/patient_home', methods=['POST'])
def patient_home():
    # Get user info
    user_id = request.args.get('user')
    patient_info = get_patient(user_id)
    data = []
    for i in list(patient_info.items()):
        data.append(str(i[0]) + ": " + str(i[1]))

    # Get message history
    message_history = get_comm_doc(user_id)
    messages = []
    for message in message_history:
        messages.append(message['message'])

    # Update graph data
    graph_type = request.form.get('graphtype')
    x, y = graph_data(user_id, graph_type)

    # Return Page
    return render_template('patient_home.html', user_info=data, messages=messages)
```

The "Method Not Allowed" (HTTP 405) here is almost certainly because the same `/patient_home` URL is being hit with a GET request somewhere, while your route only allows POST.^[1]

Main issue

Your route is:

```
@app.route('/patient_home', methods=['POST'])
def patient_home():
    ...
```

But this view is very likely also used to initially render the page (via normal navigation or a link) with a GET request, or Flask itself is redirecting to `/patient_home` via GET after a POST (for example, if you log in first, or if there is a trailing-slash redirect). Since the route only accepts POST, any GET to `/patient_home` will give “Method Not Allowed”.[\[2\]](#) [\[1\]](#)

How to fix

1. Allow GET as well, and branch on `request.method`:

```
@app.route('/patient_home', methods=['GET', 'POST'])
def patient_home():
    user_id = request.args.get('user')

    # Get user info
    patient_info = get_patient(user_id)
    data = [f"{k}: {v}" for k, v in patient_info.items()]

    # Get message history
    message_history = get_comm_doc(user_id)
    messages = [m['message'] for m in message_history]

    # Only update graph data on POST
    if request.method == 'POST':
        graph_type = request.form.get('graphtype')
        x, y = graph_data(user_id, graph_type)
    else:
        graph_type = 'weight_kg' # or whatever default
        x, y = graph_data(user_id, graph_type)

    return render_template(
        'patient_home.html',
        user_info=data,
        messages=messages,
        x_graph=x,
        y_graph=y,
        graph_type=graph_type,
    )
```

2. Ensure the form’s action is set explicitly to `/patient_home?user=...` (or uses `url_for`), so the POST goes to the same user-specific route you expect:

```
<form id="graphdataform"
      class="portal"
      method="POST"
```

```
action="{{ url_for('patient_home', user=request.args.get('user')) }}"
accept-charset="utf-8">
```

The AJAX call already uses type: "POST", url: "/patient_home", so if the page itself is reached via GET and the form/AJAX use POST, the route must support both methods.^{[1] [2]}

Extra AJAX consideration

Right now the success callback uses:

```
const x_vals = "{{ x_graph }}";
const y_vals = "{{ y_graph }}";
```

For AJAX, the server must actually return graph data in the POST response (JSON is usual), instead of a full template that was rendered earlier. A cleaner pattern is:

- Make /patient_home handle GET and render the page.
- Create a separate /patient_graph_data POST (or GET) endpoint that returns jsonify({'x': x, 'y': y}).
- In the success callback, use the JSON to build the chart.

This avoids mixing template variables with dynamic AJAX responses and reduces confusion around allowed methods.^[3]



Handle Multiple Forms

Use unique name attributes on each form's submit button. In Flask, check

request.form.get('submit_comm') (or whatever unique name) to identify which form submitted.^{[11] [12]}

HTML updates (for your graph form and others):

```
<!-- Graph form -->
<input type="submit" name="graph_submit" class="formitem" value="Send Message">

<!-- Other form example -->
<input type="submit" name="other_submit" class="formitem" value="Update Other">
```

Flask route:

```
@app.route('/patient_home', methods=['GET', 'POST'])
def patient_home():
    user_id = request.args.get('user')
    patient_info = get_patient(user_id)
    data = [{"k": k, "v": v} for k, v in patient_info.items()]
    message_history = get_comm_doc(user_id)
    messages = [{"message": m} for m in message_history]
```

```

x_graph, y_graph = None, None # defaults

if request.method == 'POST':
    if request.form.get('graph_submit'): # Graph form
        graph_type = request.form.get('graphtype')
        x_graph, y_graph = graph_data(user_id, graph_type)
    elif request.form.get('other_submit'): # Other form
        # Handle other form logic here
        pass

return render_template('patient_home.html',
                      user_info=data,
                      messages=messages,
                      x_graph=x_graph,
                      y_graph=y_graph)

```

Prevent Page Reloads (AJAX)

Your existing jQuery AJAX already prevents reloads for the graph form. Extend this pattern to all forms by handling responses as JSON fragments instead of full page reloads. [\[13\]](#)

Create dedicated JSON endpoints (cleaner than mixing with page routes):

```

@app.route('/graph_data', methods=['POST'])
def graph_data_endpoint():
    user_id = request.form.get('user')
    graph_type = request.form.get('graphtype')
    x, y = graph_data(user_id, graph_type)
    return jsonify({'x': x, 'y': y})

@app.route('/other_data', methods=['POST'])
def other_data_endpoint():
    # Return JSON for other form
    return jsonify({'status': 'updated'})

```

Updated JavaScript (attach to all forms):

```

$(document).on("submit", "#graphdataform", function(e) {
    e.preventDefault();
    $.post('/graph_data',
        {user: '{{ user_id }}', graphtype: $("#graphtype").val()},
        function(data) {
            // Update chart with data.x, data.y
            const config = {
                type: "line",
                data: {
                    labels: data.x,
                    datasets: [{ label: $("#graphtype").val(), data: data.y }]
                },
                options: { maintainAspectRatio: false }
            };
            new Chart(document.getElementById("graphdatachart"), config);
        }
    );
});

```

```

    });

    // Same pattern for other forms
    $(document).on("submit", "#otherform", function(e) {
        e.preventDefault();
        $.post('/other_data', $(this).serialize(), function(data) {
            // Update relevant page elements
            $('#other-status').text('Updated!');
        });
    });
});

```

This keeps all forms independent, AJAX-driven, and page state preserved. Pass `user_id` via hidden input or session. [\[12\]](#) [\[11\]](#) [\[13\]](#)

✱✱

It still appears to be reloading each time, and a graph never shows up:

```

    {% for info in user_info %}
{{info}}
    {% endfor %}

```

Health Tracker

...

Add observed health information and press submit. This information will be sent to your doctor.

...

Add Another Input

Enter Data

Remove Input

Health Info

{% for info in user_info %}
{{info}}

{% endfor %}

Ask Doctor

...

Send Message

...

Past Messages:

{% for message in messages %}
{{message}}

{% endfor %}

Graphical Data

Send Message

...

For Help Please Call: +1 (123) 456-7890

...

Here is app.py:

=====

Sets up the flask system to map HTML outputs to python files

By Eliot Raynor 12/02/25

=====

Import Flask and python scripts

```
from flask import Flask, redirect, render_template, request, url_for, jsonify
import os, webbrowser
from get_comm_doc import get_comm_doc
from get_comm_patients import get_comm_patients
from get_doctor_notif import get_doctor_notif
from get_patient import get_patient
from get_user_login import get_user_login
from graph_data import graph_data
```

from get_data import get_data

from submit_data import submit_data

from list_patients import list_patients

Setup Flask

```
app = Flask(name)
```

Set the default page

```
@app.route('/')
def home():
    return render_template('login_portal.html')
```

Set the login script

```
@app.route('/login', methods=['POST'])
def login():
    user = request.form['username']
    password = request.form['password']
    mode = request.form['loginmode']
    success, user_id = get_user_login(user, password, mode)

# Redirect based on login success
if success == 0:
```

```

        return redirect('/?err=1')
    elif success == 1:
        return redirect(url_for('patient_home', user=user_id))
    elif success == 2:
        return redirect(url_for('doctor_home', user=user_id))
    else:
        return redirect('/?err=2')

```

Redirect to patient page

```

@app.route('/patient_home', methods=['GET', 'POST'])
def patient_home():
    # Get user info
    user_id = request.args.get('user')
    patient_info = get_patient(user_id)
    data = [{a}: {b}] for a, b in patient_info.items()

    # Get message history
    message_history = get_comm_doc(user_id)
    messages = [m['message'] for m in message_history]

    # Return Page
    return render_template('patient_home.html', user_info=data, messages=messages)

```

Patient Form Submission

```

@app.route('/graph_data', methods=['POST'])
def graph_data_vals():
    user_id = request.form.get('user')
    graph_type = request.form.get('graphtype')
    x, y = graph_data(user_id, graph_type)
    return jsonify({'x': x, 'y': y})

@app.route('/other_data', methods=['POST'])
def other_data_endpoint():
    return jsonify({'status': 'updated'})

```

Redirect to doctor page

```

@app.route('/doctor_home')
def doctor_home():
    user_id = request.args.get('user')
    return render_template('doctor_home.html', user=user_id)

```


Solution for opening the browser from <https://stackoverflow.com/questions/54235347/open-browser-automatically-when-python-code-is-executed/54235461#54235461>

```
def open_browser():
    # webbrowser.open_new("http://127.0.0.1:5000")
    return
```

Run Flask Listener

```
if name == 'main':
    if os.environ.get('WERKZEUG_RUN_MAIN') == "true":
        open_browser()
    app.run(host='127.0.0.1', port=5000, debug=True)
```

And here is graph_data.py:

=====

Returns data from database to build graphs

By Eliot Raynor 12/07/25

=====

```
import sqlite3

def graph_data(id, graphtype=""):
    # Return if graphtype is empty
    if graphtype == "" or graphtype == None:
        return [0], [0]

id = 1 # TEMPORARY FOR DEBUGGING

# Connect to the database
conn = sqlite3.connect("NWI_DB.db")
db = conn.cursor()

# Search for x and y values
db.execute('SELECT date, %s FROM patient_visit_data WHERE id_patient=%s' %(graphtype, id))
```

```

vals = db.fetchall()
x_vals = []
y_vals = []
for point in vals:
    x_vals.append(point[0])
    y_vals.append(point[1])

# Close connection and return
conn.close()
return x_vals, y_vals

```

The main reasons the page reload happens and the graph does not show are:

1. In your AJAX success function, the chart data config object is incorrect—it lacks the actual dataset data and labels properly assigned. You wrote:

```
data: {labels: data.x, datasets: [{label: $("#graphtype").val()}]}
```

But it should include the data array inside the dataset, e.g.:

```
data: {labels: data.x, datasets: [{label: $("#graphtype").val(), data: data.y}]}
```

Without the `data: data.y`, the chart has no points to plot.

2. The JavaScript code snippet loads jQuery inside a `<script>` tag but immediately contains

3. The variable `{{ user }}` in your AJAX post data must be passed to the template context from Flask if it is used; otherwise, it's empty. You currently pass only `user_info` and `messages`. You need to pass `user=user_id` from Flask when rendering.
4. In your Flask `/graph_data` route, you use `graph_data(user_id, graph_type)` which seems correct and returns `x` and `y` arrays as JSON. That part looks okay.

Summary of fixes:

- Fix your AJAX call to properly include `data: data.y` in the chart dataset.
- Separate jQuery inclusion and your JavaScript code into two `<script>` tags.
- Pass `user=user_id` from Flask `patient_home` route for template usage.
- Make sure your canvas element exists and is properly sized.

Example corrected snippet from your HTML:

```

<script src="https://code.jquery.com/jquery-3.5.1.js"></script>
<script type="text/javascript">
    $(document).on("submit", "#graphdataform", function(e) {
        e.preventDefault();
        $.post("/graph_data",
            {user: "{{ user }}", graphtype: $("#graphtype").val()},

```

```

        function(data) {
            const config = {
                type: "line",
                data: {
                    labels: data.x,
                    datasets: [{
                        label: $("#graphtype").val(),
                        data: data.y,
                        borderColor: 'rgba(75, 192, 192, 1)',
                        fill: false
                    }]
                },
                options: { maintainAspectRatio: false }
            };
            new Chart(document.getElementById("graphdatachart"), config);
        });
    </script>

```

And from your Flask route for patient_home:

```

@app.route('/patient_home', methods=['GET', 'POST'])
def patient_home():
    user_id = request.args.get('user')
    patient_info = get_patient(user_id)
    data = [f'{a}: {b}' for a, b in patient_info.items()]
    message_history = get_comm_doc(user_id)
    messages = [m['message'] for m in message_history]

    return render_template('patient_home.html', user_info=data, messages=messages, user=

```

These changes will prevent the page reload on form submit and show the graph correctly by receiving and plotting the proper data without full page refresh. Also confirm the Chart.js library is included in your HTML (not shown in your snippet but required for charts to render).

Let me know if you want precise edits for the full HTML or Flask code with this incorporated.

Fix JavaScript Errors

Two critical issues cause these errors: **timing** (scripts run before DOM elements exist) and **jQuery loading**.^{[21] [22]}

1. document.getElementById(...) is null at line 129

This is in the **Ask Doctor form**:

```

<input type="hidden" name="date_comm" value="">
<script> document.getElementById("date_comm").value = date; </script>

```

Problems:

- Script runs before `<input id="date_comm">` exists (line 129)
- `date` variable is undefined (defined elsewhere)

Fix:

```
<!-- Ask Doctor form - FIXED -->
<form id="askdoctorform" class="portal" method="POST" accept-charset="utf-8">
  <input type="hidden" name="date_comm" id="date_comm" value="">
  <input type="text" name="comm_text" class="formitem" id="commtext" value="">
  <input type="submit" name="submit_comm" class="formitem" value="Send Message">
</form>

<script>
// Set date after DOM loads
document.addEventListener('DOMContentLoaded', function() {
  const now = new Date().toJSON();
  document.getElementById('date_comm').value = now;
});
</script>
```

2. \$ is not defined at line 166 (Graph form)

jQuery script tag is malformed - you have code *inside* the `src` tag:

```
<!-- BROKEN -->
<script src="https://code.jquery.com/jquery-3.5.1.js">
  // Your AJAX code here - IGNORED BY BROWSER
</script>
```

Fix - Separate tags and move to page bottom:

```
<!-- At end of Graphical Data section, BEFORE closing </div> -->
<script src="https://code.jquery.com/jquery-3.5.1.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!-- ADD Chart.js -->
<script>
$(document).ready(function() { // Wait for jQuery + DOM
  $(document).on("submit", "#graphdataform", function(e) {
    e.preventDefault();
    $.post("/graph_data",
      {user: "{{ user }}", graphtype: $("#graphtype").val()},
      function(data) {
        const ctx = document.getElementById("graphdatachart").getContext('2d')
        new Chart(ctx, {
          type: "line",
          data: {
            labels: data.x,
            datasets: [{
              label: $("#graphtype").val(),
              data: data.y,
              borderColor: 'rgba(75, 192, 192, 1)',
              fill: false
            }]
          }
        })
      })
  })
})
```

```

        },
        options: { maintainAspectRatio: false }
    });
});
});
});
</script>

```

3. Flask template fix

Add user=user_id to your patient_home route:

```

return render_template('patient_home.html',
    user_info=data,
    messages=messages,
    user=user_id) # ADD THIS

```

4. Move ALL scripts to bottom of <body>

Put jQuery, Chart.js, and all custom scripts in a single <script> block just before </body>

These fixes resolve both errors and make AJAX work without page reloads.

✱✱

22:09:21.730 Uncaught ReferenceError: \$ is not defined

<anonymous> http://127.0.0.1:5000/patient_home?user=0:171

[patient_home:171:21](#)

22:09:21.746 Loading failed for the <script> with source "<https://code.jquery.com/jquery-3.5.1.js>". [patient_home:203:63](#)

22:09:21.803 Loading failed for the <script> with source "<https://cdn.jsdelivr.net/npm/chart.js>". [patient_home:204:61](#)

22:09:21.808

Uncaught TypeError: can't access property "value", document.getElementById(...) is null

<anonymous> http://127.0.0.1:5000/patient_home?user=0:132

EventListener.handleEvent* http://127.0.0.1:5000/patient_home?user=0:130

[patient_home:132:46](#)

22:09:21.851

GET <http://127.0.0.1:5000/favicon.ico>[HTTP/1.1 404 NOT FOUND 0ms]

```

    {% for info in user_info %}
    {{info}}

```

{% endfor %}

Health Tracker

...

Add observed health information and press submit. This information will be sent to your doctor.

...

Add Another Input

Enter Data

Remove Input

Health Info

{% for info in user_info %}

{{info}}

{% endfor %}

Ask Doctor

Send Message

Past Messages:

{% for message in messages %}

{{message}}

{% endfor %}

Graphical Data

Send Message

...

For Help Please Call: +1 (123) 456-7890

...

... ..

Two separate things are happening:

1. jQuery and Chart.js are not loading at all (so \$ is undefined).
2. The `date_comm` hidden input is missing an `id`, so `getElementById("date_comm")` returns `null`.

Below is the minimal set of changes to get rid of the errors and make the graph update without reloading.

1. Fix \$ is not defined (CDN load failure)

Your console shows that both script loads failed, so all code that uses \$ or Chart will fail. The two most common reasons are:

- No internet access from the machine running the browser (or CDN blocked).
- A CSP / firewall / offline environment blocking external scripts. [\[30\]](#) [\[31\]](#)

To make this robust, download the JS files and serve them from your Flask `static` folder instead of CDNs:

1. Download:

- `jquery-3.5.1.min.js`
- `chart.umd.min.js` (from the Chart.js release)

2. Put them in `static/js/`:

- `static/js/jquery-3.5.1.min.js`
- `static/js/chart.umd.min.js`

3. In your template, replace the CDN tags at the bottom with:

```
<script src="{{ url_for('static', filename='js/jquery-3.5.1.min.js') }}"></script>
<script src="{{ url_for('static', filename='js/chart.umd.min.js') }}"></script>
```

4. Move your graph code **after** those two tags, so \$ and Chart exist before you use them:

```
<script>
$(document).ready(function() {
    $(document).on("submit", "#graphdataform", function(e) {
        e.preventDefault();
        $.post("/graph_data",
            {user: "{{ user }}"}, graphtype: $("#graphtype").val()),
        function(data) {
            const ctx = document.getElementById("graphdatachart").getContext('2d');
            new Chart(ctx, {
                type: "line",
                data: {
                    labels: data.x,
                    datasets: [{
                        label: $("#graphtype").val(),
                        data: data.y,
                        borderColor: 'rgba(75, 192, 192, 1)',
                        fill: false
                    }]
                },
                options: { maintainAspectRatio: false }
            });
        }
    });
});
</script>
```

Also ensure that your Flask `patient_home` route passes a `user` variable (not `user_id`, since your template uses `{{ user }}`):

```
return render_template(
    'patient_home.html',
    user_info=data,
    messages=messages,
    user=user_id
)
```

This removes the `$ is not defined` error as long as the browser can reach your Flask static files.
[\[30\]](#)

2. Fix `document.getElementById(...)` is null **for** `date_comm`

In your Ask Doctor form you have:

```
<input type="hidden" name="date_comm" value="">
<script>
    document.addEventListener("DOMContentLoaded", function() {
        const date2 = new Date().toJSON();
        document.getElementById("date_comm").value = date2;
    });
</script>
```



```
});  
</script>
```

The input has **no** `id="date_comm"`, so `getElementById("date_comm")` returns `null`.^[31] ^[32]

Change that block to:

```
<form id="askdoctorform" class="portal" action="/login" method="POST" accept-charset="utf-8">  
  <input type="hidden" name="date_comm" id="date_comm" value="">  
  <input type="text" name="comm_text" class="formitem" id="commtext" value="">  
  <input type="submit" name="submit_comm" class="formitem" value="Send Message">  
</form>  
  
<script>  
document.addEventListener("DOMContentLoaded", function() {  
  const date2 = new Date().toJSON();  
  const el = document.getElementById("date_comm");  
  if (el) {  
    el.value = date2;  
  }  
});  
</script>
```

This ensures the element exists and is found correctly.^[32]

3. Small consistency fix in graph JS

In the snippet you posted you use `user: "{{ user_id }}"`, but your Flask code uses `user_id` only on the Python side. To match the template variable you pass, either:

- Pass `user_id` as `user_id` and use `{{ user_id }}` in the template; or
- Pass `user` and use `{{ user }}` consistently.

Pick one and keep it the same in `render_template`, in your HTML, and in your AJAX data.^[30]

Once:

- jQuery and Chart.js load from your `static` folder,
- The `date_comm` input has an `id`,
- The `user` variable is consistent,

the errors will disappear and the graph POST should fire without reloading the page.



Destroy Chart Before Reusing Canvas

Chart.js requires destroying the existing chart instance before creating a new one on the same canvas. Store the chart reference globally and call `.destroy()` before recreating. [\[33\]](#) [\[34\]](#)

Replace your graph JavaScript with this:

```
<script>
let graphChart = null; // Global chart reference

$(document).ready(function() {
  $(document).on("submit", "#graphdataform", function(e) {
    e.preventDefault();

    // Destroy existing chart if it exists
    if (graphChart) {
      graphChart.destroy();
      graphChart = null;
    }

    $.post("/graph_data",
      {user: "{{ user }}", graphtype: $("#graphtype").val()},
      function(data) {
        const canvas = document.getElementById("graphdatachart");
        const ctx = canvas.getContext('2d');

        graphChart = new Chart(ctx, {
          type: "line",
          data: {
            labels: data.x,
            datasets: [{
              label: $("#graphtype").val(),
              data: data.y,
              borderColor: 'rgba(75, 192, 192, 1)',
              backgroundColor: 'rgba(75, 192, 192, 0.2)',
              fill: false,
              tension: 0.1
            }]
          },
          options: {
            maintainAspectRatio: false,
            scales: {
              y: { beginAtZero: true }
            }
          }
        });
      }
    );
  });
});
</script>
```

Key changes:

1. `let graphChart = null;` - stores the chart instance globally

2. `if (graphChart) { graphChart.destroy(); graphChart = null; }` - destroys before recreating ^[33]
3. Added `tension: 0.1` for smoother lines and `beginAtZero: true` for better y-axis scaling

Place this script AFTER the jQuery and Chart.js script tags (at bottom of `<body>`). ^[33]

This eliminates the "Canvas is already in use" error completely. The chart will update smoothly each time you change the graph type without page reloads. ^[34]

✱✱

1. <https://stackoverflow.com/questions/21689364/method-not-allowed-flask-error-405>
2. https://www.reddit.com/r/flask/comments/ljpxhx/how_to_solve_405_method_not_allowed_flask/
3. <https://www.youtube.com/watch?v=MrK8fsiUPes>
4. https://www.reddit.com/r/flask/comments/pa32yg/i_am_unable_to_route_my_app_using_flask_i_keep/
5. https://www.reddit.com/r/flask/comments/lfydd2/405_method_not_allowed/
6. <https://community.postman.com/t/flask-the-method-is-not-allowed-for-the-requested-url/20086>
7. https://www.reddit.com/r/flask/comments/4fnler/ask_flask_unsupported_method_post_by_simple/
8. <https://stackoverflow.com/questions/54726092/flask-and-python-method-not-allowed-while-trying-to-read-from-form>
9. <https://colalove5562.tistory.com/72>
10. https://www.reddit.com/r/flask/comments/fnwk0h/405_method_not_allowed_in_flask/
11. https://www.youtube.com/watch?v=j0iH6N_9wNg
12. <https://stackoverflow.com/questions/18290142/multiple-forms-in-a-single-page-using-flask-and-wtforms>
13. https://www.reddit.com/r/flask/comments/y54nur/how_do_you_submit_multiple_forms_at_once/
14. https://www.reddit.com/r/flask/comments/9bu1k1/flask_handle_multiple_forms/
15. <https://www.youtube.com/watch?v=4aijeclGG9A>
16. <https://stackoverflow.com/questions/68469133/multiple-forms-in-one-page-flask>
17. https://www.reddit.com/r/flask/comments/tbwlnm/multiple_forms_in_the_same_page_with_flaskwtforms/
18. <https://dev.to/sampart/combining-multiple-forms-in-flask-wtforms-but-validating-independently-cbm>
19. <https://teamtreehouse.com/community/multiple-forms-in-a-single-route-page-with-wtforms>
20. https://www.reddit.com/r/flask/comments/gsrwsy/is_it_possible_to_have_more_than_one_of_a_single/
21. <https://stackoverflow.com/questions/8739605/getelementbyid-returns-null>
22. <https://stackoverflow.com/questions/2632137/why-is-document-getelementbyid-returning-null>
23. <https://www.sololearn.com/en/Discuss/1143121/javascript-getelementbyid-keeps-returning-null>
24. https://www.reddit.com/r/learnprogramming/comments/104khpu/what_am_i_doing_wrong_uncaught_typeerror/
25. https://www.reddit.com/r/learnjavascript/comments/21lj7t/getelementbyid_returns_null/
26. <https://forum.freecodecamp.org/t/typeerror-document-getelementbyid-is-null/450883>
27. <https://terracoders.com/node/76>

28. <https://forum.webix.com/t/getelementbyid-is-returning-returning-null/15340>
29. <https://teamtreehouse.com/community/cant-figure-out-why-i-am-getting-null-when-i-getelementbyid-here-would-really-appreciate-help>
30. <https://stackoverflow.com/questions/8739605/getelementbyid-returns-null>
31. <https://forum.freecodecamp.org/t/typeerror-document-getelementbyid-is-null/450883>
32. <https://stackoverflow.com/questions/2632137/why-is-document-getelementbyid-returning-null>
33. <https://stackoverflow.com/questions/40056555/destroy-chart-js-bar-graph-to-redraw-other-graph-in-same-canvas>
34. <https://www.chartjs.org/docs/latest/developers/api.html>
35. <https://blog.csdn.net/zhoupenghui168/article/details/134011282>
36. <https://stackoverflow.com/questions/68100325/canvas-is-already-in-use-chart-with-id-0-must-be-destroyed-before-the-canvas/68407708>
37. <https://github.com/reactchartjs/react-chartjs-2/issues/1037>
38. <https://github.com/chartjs/Chart.js/issues/8002>
39. <https://www.geeksforgeeks.org/javascript/how-to-clear-chart-from-canvas-and-disable-hover-events/>
40. <https://www.chartjs.org/docs/latest/developers/api.html?h=destroy>
41. <https://github.com/reactchartjs/react-chartjs-2/issues/675>
42. <https://github.com/chartjs/Chart.js/issues/1007>