LAB #3: WEB APPLICATION WITH GENIE

Mohamed Habib Ouadhour
Dept. of EE (AII21)
ISET Bizerte — Tunisia
ouadhourhabib@yahoo.com

I. INTRODUCTION

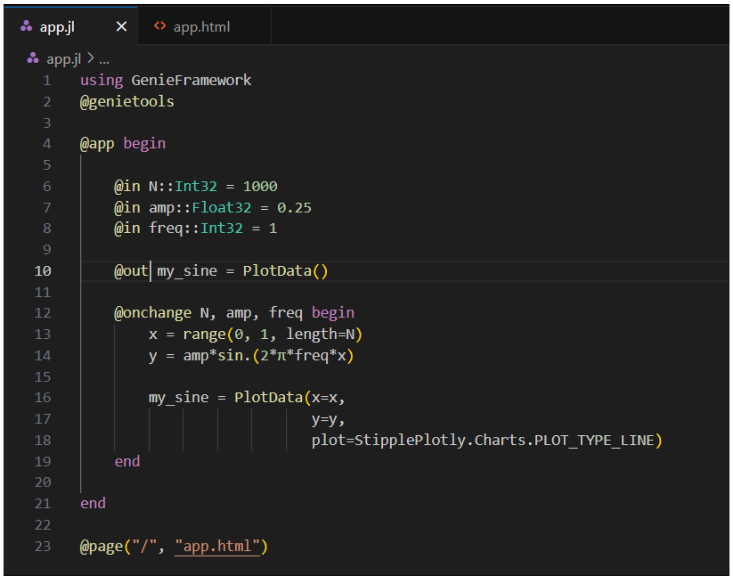
In this lab, I created a basic web application using **Genie** framework in Julia. This application will allow us to control the behaviour of a sine wave, given some adjustble parameters.

Figure 1: Genie

As we know to build and run a web application we need a Julia and HTML codes.

1. **Testing Codes :**

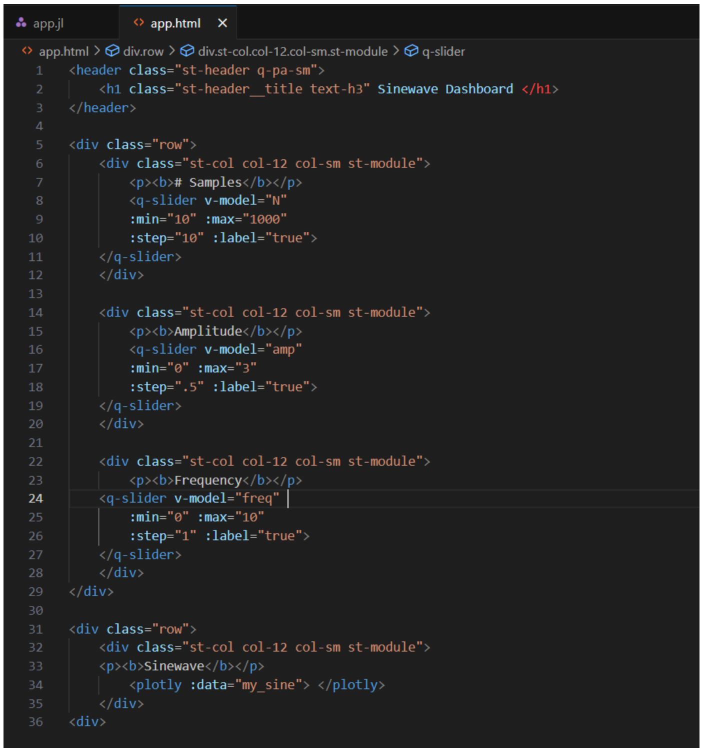
- Julia :



```julia
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1

    @out my_sine = PlotData()

    @onchange N, amp, freq begin
        x = range(0, 1, length=N)
        y = amp*sin.(2*π*freq*x)

        my_sine = PlotData(x=x,
                           y=y,
                           plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
    end

end

@page("/", "app.html")
```

Figure 3: Code of Julia

- HTML :

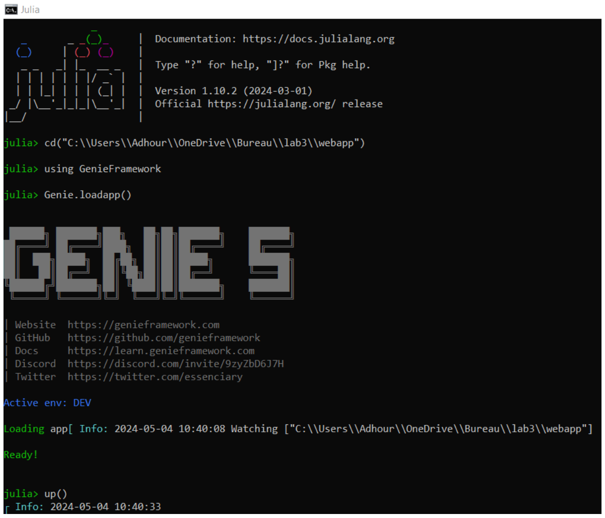


```html
<header class="st-header q-pa-sm">
    <h1 class="st-header__title text-h3" Sinewave Dashboard </h1>
</header>

<div class="row">
    <div class="st-col col-12 col-sm st-module">
        <p><b># Samples</b></p>
        <q-slider v-model="N"
        :min="10" :max="1000"
        :step="10" :label="true">
    </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>Amplitude</b></p>
        <q-slider v-model="amp"
        :min="0" :max="3"
        :step=".5" :label="true">
    </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>Frequency</b></p>
    <q-slider v-model="freq"
        :min="0" :max="10"
        :step="1" :label="true">
    </q-slider>
    </div>
</div>

<div class="row">
    <div class="st-col col-12 col-sm st-module">
    <p><b>Sinewave</b></p>
        <plotly :data="my_sine"> </plotly>
    </div>
<div>
```

Figure 4: Code of HTML

- Result :

```
julia> using GenieFramework
julia> Genie.loadapp() # Load app
julia> up() # Start server
```



```
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.10.2 (2024-03-01)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> cd("C:\\Users\\Adhour\\OneDrive\\Bureau\\lab3\\webapp")

julia> using GenieFramework

julia> Genie.loadapp()

GENIE 5

| Website  https://genieframework.com
| GitHub   https://github.com/genieframework
| Docs     https://learn.genieframework.com
| Discord  https://discord.com/invite/9zyZbD6J7H
| Twitter  https://twitter.com/essenciary

Active env: DEV

Loading app[ Info: 2024-05-04 10:40:08 Watching ["C:\\Users\\Adhour\\OneDrive\\Bureau\\lab3\\webapp"]

Ready!


julia> up()
┌ Info: 2024-05-04 10:40:33
```

Figure 5: Julia REPL

We can now open the browser and navigate to the link localhost:8000. We will get the graphical interface as in *Figure 6*.

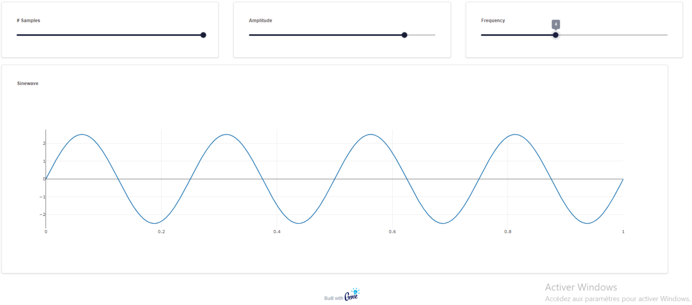


Figure 6: Graphical interface

II. Exercices

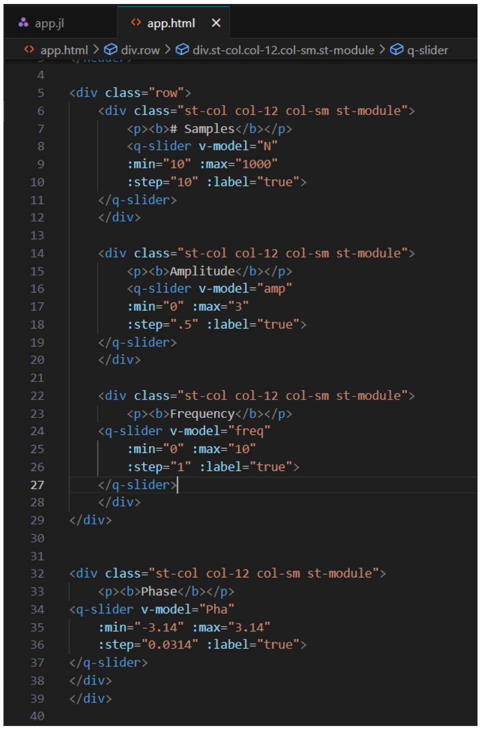
As we know the mathematique equation of sin wave is:

$$v(t) = V_M.\sin(\omega.t + \varphi)$$

So I need to complete the missing variables: Phase and Offset

- **First task:**

In the first task, I added a slide that modify the *Phase* **ranging between $-\pi$ and π, changes by a step of $\frac{\pi}{100}$**

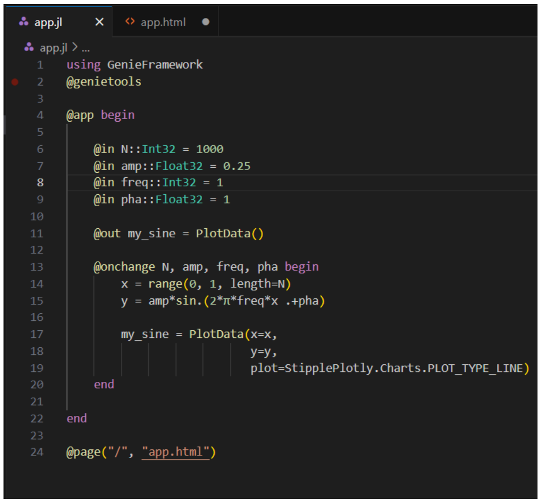


app.jl | app.html ×

app.html > div.row > div.st-col.col-12.col-sm.st-module > q-slider

```html

<div class="row">
    <div class="st-col col-12 col-sm st-module">
        <p><b># Samples</b></p>
        <q-slider v-model="N"
        :min="10" :max="1000"
        :step="10" :label="true">
    </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>Amplitude</b></p>
        <q-slider v-model="amp"
        :min="0" :max="3"
        :step=".5" :label="true">
    </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>Frequency</b></p>
    <q-slider v-model="freq"
        :min="0" :max="10"
        :step="1" :label="true">
    </q-slider>
    </div>
</div>


<div class="st-col col-12 col-sm st-module">
    <p><b>Phase</b></p>
<q-slider v-model="Pha"
    :min="-3.14" :max="3.14"
    :step="0.0314" :label="true">
</q-slider>
</div>
</div>

```

Figure 7: Adding slide for Phase



app.jl × | app.html ●

app.jl > ...

```julia
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1
    @in pha::Float32 = 1

    @out my_sine = PlotData()

    @onchange N, amp, freq, pha begin
        x = range(0, 1, length=N)
        y = amp*sin.(2*π*freq*x .+pha)

        my_sine = PlotData(x=x,
                           y=y,
                           plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
    end

end

@page("/", "app.html")
```

Figure 8: Adding the phase function in Julia

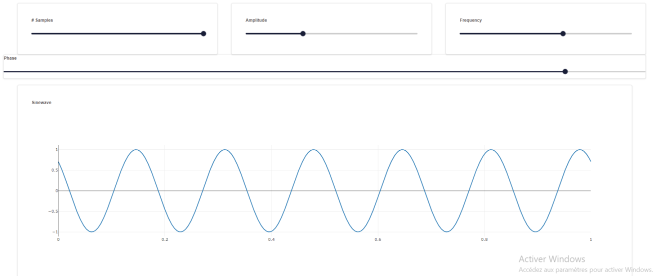
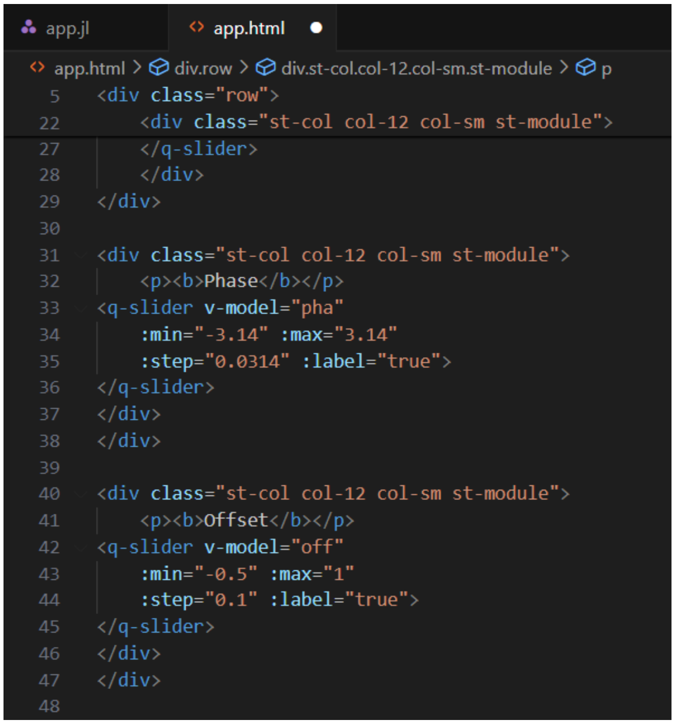


Figure 9: Graphical Interface

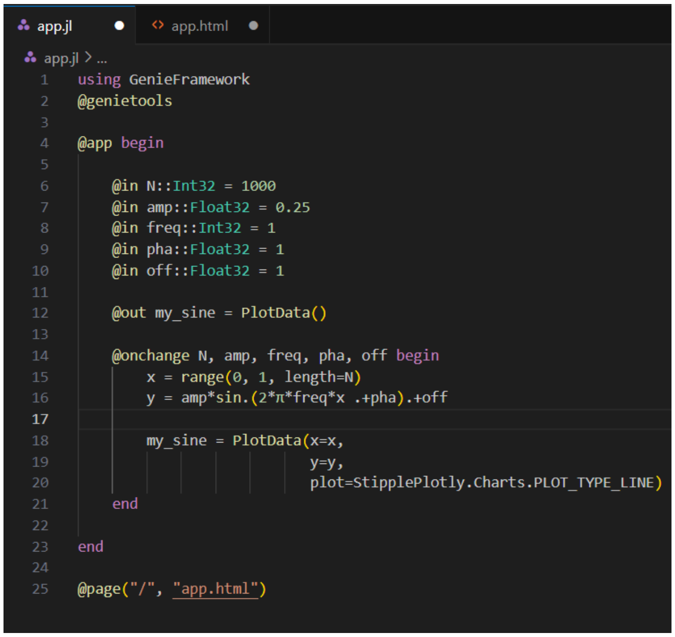
- **Second task:**

Then in the second task, I added a slide that modify the *Offset* **varies from** -0.5 **to** 1**, by a step of** 0.1**.**



```html
<div class="row">
    <div class="st-col col-12 col-sm st-module">
    </q-slider>
    </div>
</div>

<div class="st-col col-12 col-sm st-module">
    <p><b>Phase</b></p>
<q-slider v-model="pha"
    :min="-3.14" :max="3.14"
    :step="0.0314" :label="true">
</q-slider>
</div>
</div>

<div class="st-col col-12 col-sm st-module">
    <p><b>Offset</b></p>
<q-slider v-model="off"
    :min="-0.5" :max="1"
    :step="0.1" :label="true">
</q-slider>
</div>
</div>
```

Figure 10: Adding Slide for Offset



```julia
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1
    @in pha::Float32 = 1
    @in off::Float32 = 1

    @out my_sine = PlotData()

    @onchange N, amp, freq, pha, off begin
        x = range(0, 1, length=N)
        y = amp*sin.(2*π*freq*x .+pha).+off

        my_sine = PlotData(x=x,
                           y=y,
                           plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
    end

end

@page("/", "app.html")
```

Figure 11: Adding the offset function in Julia

-> Final result of the graphical interface with all the sin wave variables (*Figure 12*).

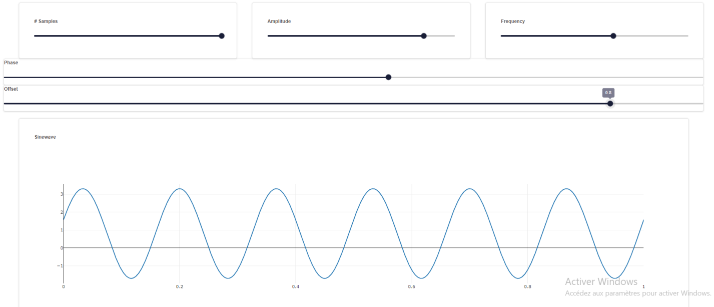


Figure 12: Final Graphical Interface

III. Conclusion

This lab permit me to learn how to create a web application using Genie in Julia.

References