Lab 3: Web Application with Genie

Mohamed Raed Boukari
Dept. of EE (AII21)
ISET Bizerte — Tunisia
raedboukari2018@gmail.com

Sana Ben Said
Dept. of EE (AII21)
ISET Bizerte — Tunisia
bensaidsana06@gmail.com

I. Introduction

In this Lab we used Genie Framework in julia , Genie is a full-stack MVC web framework that provides a streamlined and efficient workflow for developing modern web applications. It builds on Julia's strengths (high-level, high-performance, dynamic, JIT compiled), exposing a rich API and a powerful toolset for productive web development.

II. Exercices

- In the first task we add extra slide that modify the behaviour of the sine wave graph by adding : ***Phase* ranging between $-\pi$ and π, changes by a step of $\frac{\pi}{100}$**

```julia
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1
    @in ph::Float64 = 3.14/100


    @out my_sine = PlotData()

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

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

end

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

Figure 1: Adding the phase function in julia

```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>phase</b></p>
        <q-slider v-model="ph"
        :min="-3.14" :max="3.14"
        :step="3.14/100" :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 2: Adding the phase function in HTML

- Then in the second task we add the offset function : **Offset varies from -0.5 to 1, by a step of 0.1**

```julia
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1
    @in ph::Float64 = 3.14/100
    @in offset::Int32 = 0


    @out my_sine = PlotData()

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

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

end

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

Figure 3: Adding the offset function in Julia

```html
<div class="row">
    <div class="st-col col-12 col-sm st-module">
        <q-slider v-model="N"
        :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>phase</b></p>
        <q-slider v-model="ph"
        :min="-3.14" :max="3.14"
        :step="3.14/100" :label="true">
    </q-slider>
    </div>
    <div class="st-col col-12 col-sm st-module">
        <p><b>offset</b></p>
        <q-slider v-model="offset"
        :min="-0.5" :max="1"
        :step="0.1" :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: Adding the offset function in HTML

- After that we open the Julia REPL in order to open the web app

Figure 5: Julia REPL

- Now , we have the sine wave and we can change every parameter thanks to the graphic interface

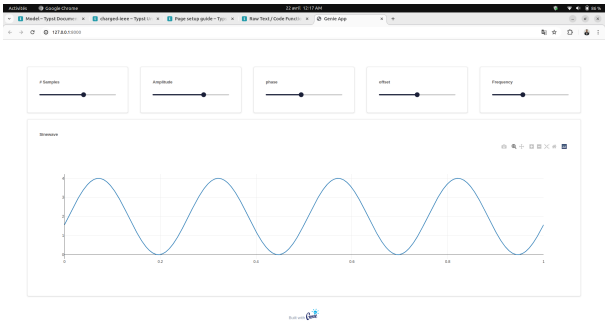

Figure 6: Sine Wave

III. Conclusion

In this Lab we have the ability to use Genie Lab in julia to design a mathematical web app

References