

User/PieChart

.module.exports() => (**HTML**)

Displays user stats, regarding their performance in quizzes.

Kind: static method of **User/PieChart**

Code

```
/**
 * @module User/PieChart
 * @memberof User
 */

import { Chart as ChartJS, ArcElement, Tooltip, Legend } from 'chart.js';
import { Pie } from 'react-chartjs-2';
import { useEffect, useState } from "react";

/**
 * Displays user stats, regarding their performance in quizzes.
 * @function
 * @return {HTML}
 */
export default function PieChart(){

  /**
   * Stores how many quizzes a user has done correctly, and a function to change the
   value of "done".
   * @constant
   * @type {Array.<String, Function>}
   * @property {}
   */
  const [done, set_done] = useState(5)

  /**
   * Stores how many quizzes a user has attempted, and a function to change the value
   of "attempted".
   * @constant
   * @type {Array.<String, Function>}
   * @param
   */
  const [attempted, set_attempted] = useState(7)

  /**
   * Stores how many quizzes a user has remaining, and a function to change the value
   of "remaining".
   * @constant
   * @type {Array.<String, Function>}
```

```

*/
const [remaining, set_remaining] = useState(5)

useEffect(
  ()=>{
    console.log(localStorage.getItem("id"))
    fetch("http://localhost:8000/user/quiz/stats/overall",{
      method: "POST",
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(
        {
          "student": true,
          "id": localStorage.getItem("id")
        }
      )
    })
    .then(response => response.json())
    .then(data => {
      const quiz_stats = data
      set_done(quiz_stats["done"])
      set_attempted(quiz_stats["attempted"])
      set_remaining(quiz_stats["remaining"])
    })
  }, []
)

ChartJS.register(ArcElement, Tooltip, Legend);

/**
 * Stores labels for the Pie chart.
 * @constant
 * @property {string} "Quizes done"
 * @property {string} "Quizes attempted"
 * @property {string} "Quizes remaining"
 */
const labels = [
  "Quizes done",
  "Quizes attempted",
  "Questions remaining"
]

/**
 * @constant
 * @property {Object} labels.labels
 * @property {Object.Array} datasets[0]
 * @property {string} datasets[0].label
 * @property {Object.Array} "(datasets[0].data[done] |
datasets[0].data[attempted] | datasets[0].data[remaining])"

```

```

    * @property {Object.Array} "(datasets[0].backgroundColor[0] |
    datasets[0].backgroundColor[1] | datasets[0].backgroundColor[2])"
    * @property {Object.Array} "(datasets[0].borderColor[0] |
    datasets[0].borderColor[1] | datasets[0].borderColor[2])"
    * @property {integet} datasets[0].borderWidth
    * @property {string} datasets[0].label
  */
  const data = {
    labels: labels,
    datasets: [{
      label: '',
      data: [done, attempted, remaining],
      backgroundColor: [
        '#569D1B',
        '#CC350B',
        '#004586',
      ],
      borderColor: [
        '#569D1B',
        '#CC350B',
        '#FFD320',
      ],
      borderWidth: 1
    }]
  };

  /**
  * @constant
  * @property {Object} plugins
  * @property {Object} legend
  * @property {string} plugins.legend.position
  * @property {string} plugins.legend.align
  * @property {Object} plugins.legend.labels
  * @property {Object} plugins.legend.labels.font
  * @property {String} plugins.legend.labels.font.size
  */
  const options = {
    plugins:{
      legend: {
        position: "right",
        align: "middle",
        labels:{
          font:{
            size: "20"
          }
        }
      }
    }
  }
}

```

```
    return(  
      <Pie className="pie-chart" data={data} options={options} height="50"  
width="50" />  
    )  
  }
```