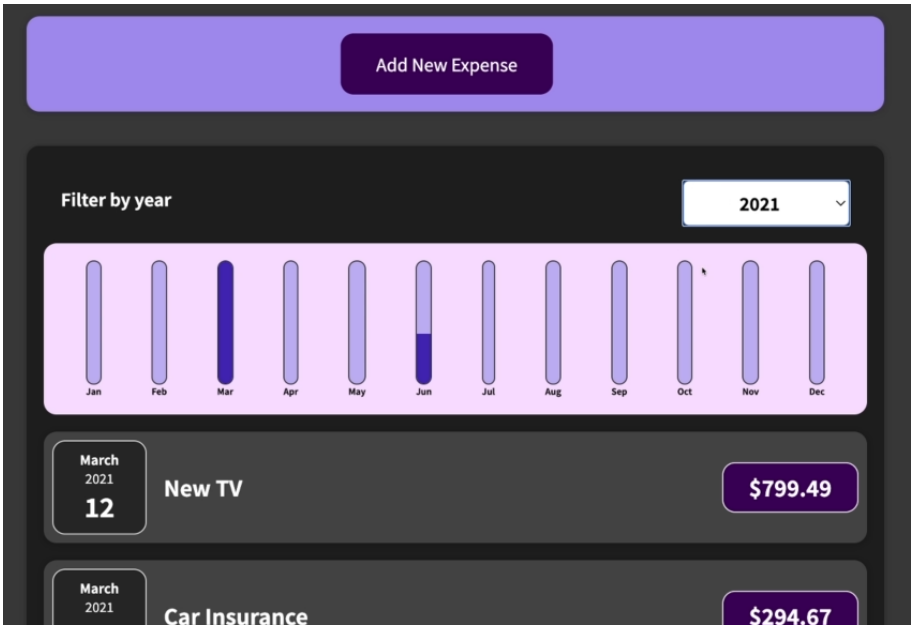
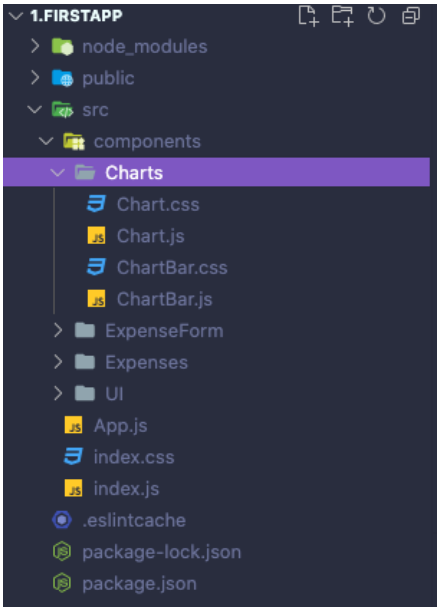
32. Adding dynamic styling and chartbar component of Expense tracker project

- Aim:
  - 

- Now let us create a new component folder for charts
  - 

- Here Chart.js contains all the bars(i.e bars for every month)
- ChartBar.js contains the specific Bar(i.e single bar component)

- In Chart.js
  - src > components > Charts > Chart.js > Chart

```
import React from "react";
import './Chart.css';
import './ChartBar';

const Chart=()=>{
    const chartDataPoints = [
        { label: 'Jan', value: 0 },
        { label: 'Feb', value: 0 },
        { label: 'Mar', value: 0 },
        { label: 'Apr', value: 0 },
        { label: 'May', value: 0 },
        { label: 'Jun', value: 0 },
        { label: 'Jul', value: 0 },
        { label: 'Aug', value: 0 },
        { label: 'Sep', value: 0 },
        { label: 'Oct', value: 0 },
        { label: 'Nov', value: 0 },
        { label: 'Dec', value: 0 },
      ];

    return(
        <div className="chart">
        {chartDataPoints.map((dataPoint)=>{<ChartBar label={dataPoint.label} value={dataPoint.value}></ChartBar>})
        </div>
    )
}
export default Chart;
```

    - Here we have datapoint for each month, let us use map function to render chartbar for all the 12 months along with passing the value and the label
- In chartbar.js
  - src > components > Charts > ChartBar.js > ChartBar

```
import React from 'react';

import './ChartBar.css';

const ChartBar=(props)=>{
    let barFillHeight='40%';

    return(
        <div className='chart-bar'>
            <div className='chart-bar__capsule'>
                <div className='chart-bar__fill' style={}></div>
            </div>
            <div className='chart-bar__label'></div>

        </div>
    )
}

export default ChartBar;
```

- Here in line 6 we have declared an variable which will be changed for each month later so we used let keyword, we will be making it dynamic for each month using some calculations and logic later
- **In line 11, we are going to pass the dynamic style through JSX code**

src > components > Charts > ChartBar.js > ChartBar

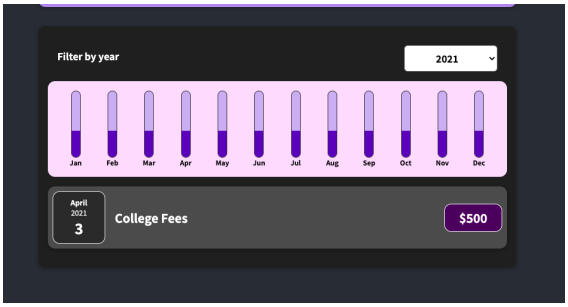
```
5   const ChartBar=(props)=>{
6       let barFillHeight='40%';
7
8       return(
9           <div className='chart-bar'>
10              <div className='chart-bar__capsule'>
11                  <div className='chart-bar__fill' style={{height:barFillHeight}}></div>
12              </div>
13              <div className='chart-bar__label'>{props.label}</div>
14
15          </div>
16      )
17  }
18
19  export default ChartBar;
```

- **<u>Here we should pass the styles only in objects</u>**
- e.g→style={{height:'40%','background-color':'#ffff'}}
  - Here for background-color we are enclosing it with quotes because there we have '-' character so it must be enclosed within single quotes

- In Expenses.js

src > components > Expenses > Expenses.js > Expenses

```
18
19
20
21      return (
22
23        <Card className="expenses">
24          <ExpenseFilter onChangeFilter={changeFilterhandler}
25          <Chart></Chart>
26          <ExpensesList items={filteredExpenses} ></ExpensesL
27        </Card>
28      );
29    }
30
31    export default Expenses;
```

- This gives this output
  - 

    - This is because we have set the height of the fill as 40%(static)
- Now its time to make some calculation for each month and find out the percentage
- In expenses.js
  - ```
    src > components > Expenses > Expenses.js > Expenses
    21      return (
    22
    23        <Card className="expenses">
    24          <ExpenseFilter onChangeFilter={changeFilterhandler} selected={filteredYear} ></ExpenseFilter>
    25          <Chart expenses={filteredExpenses}></Chart>
    26          <ExpensesList items={filteredExpenses} ></ExpensesList>
    27        </Card>
    28      );
    29    }
    30
    31    export default Expenses;
    ```
    - Here we are passing the filteredExpenses to Chart Component
- In chart.js
  - ```
    src > components > Charts > Chart.js > Chart
    5   const Chart=(props)=>{
    6       const chartDataPoints = [
    7           { label: 'Jan', value: 0 },
    8           { label: 'Feb', value: 0 },
    9           { label: 'Mar', value: 0 },
    10          { label: 'Apr', value: 0 },
    11          { label: 'May', value: 0 },
    12          { label: 'Jun', value: 0 },
    13          { label: 'Jul', value: 0 },
    14          { label: 'Aug', value: 0 },
    15          { label: 'Sep', value: 0 },
    16          { label: 'Oct', value: 0 },
    17          { label: 'Nov', value: 0 },
    18          { label: 'Dec', value: 0 },
    19        ];
    20
    21        for(const expense of props.expenses){
    22            const expenseMonthIndex=expense.date.getMonth(); //this return index 0-11 respectively extracting the mo
    23            chartDataPoints[expenseMonthIndex].value+=expense.amount;
    24        }
    25
    26        const dataPointValuesList=chartDataPoints.map((dataPoint)=>{return dataPoint.value})
    27        const maxValueComparingAllMonths=Math.max(...dataPointValuesList);
    28
    29      return(
    30          <div className="chart">
    31          {chartDataPoints.map((dataPoint)=>(
    32              <ChartBar label={dataPoint.label} value={dataPoint.value}></ChartBar>
    ```

- Here in line 21,we are looping through the filteredexpenses that is passed through props, and getting the month index using getMonth() and we are using that index value which is used to update the chartDatapoints list by adding the expenseamount to respective month
- In line 27, we are getting the maximum value comparing all the months and selecting the month with the highest expense
  - In line 26, we are creating a list which will hold only the values from the chartdatapoints list
  - In line 27, we are destructuring the values list and finding the maximum value
  - Now we have to pass it to chartBar component so that we can make calculations there
  -

```
src > components > Charts > Chart.js > Chart > chartDataPoints.map() callback
21        for(const expense of props.expenses){
22            const expenseMonthIndex=expense.date.getMonth(); //this return index 0-11 respectively extracting the mo
23            chartDataPoints[expenseMonthIndex].value+=expense.amount;
24        }
25
26        const dataPointValuesList=chartDataPoints.map((dataPoint)=>{return dataPoint.value})
27        const maxValueComparingAllMonths=Math.max(...dataPointValuesList);
28
29      return(
30          <div className="chart">
31          {chartDataPoints.map((dataPoint)=>(
32              <ChartBar label={dataPoint.label} value={dataPoint.value} maxValue={maxValueComparingAllMonths}></Char
33              )
34              )
35          }
36          </div>
37      )
38  }
39  export default Chart;
```

- In ChartBar.js
  -

```
src > components > Charts > ChartBar.js > ChartBar
1   import React from 'react';
2
3   import './ChartBar.css';
4
5   const ChartBar=(props)=>{
6       let barFillHeight='0%';
7
8       if (props.maxValue>0){
9           barFillHeight=Math.round((props.value/props.maxValue)*100)+'%' //here adding '%'
10      }
11
12      return(
13          <div className='chart-bar'>
14              <div className='chart-bar__capsule'>
```

  - Here we are calculating
  - Let us consider the maxvalue as 100 percent, so we are finding percentage of other values based on the maxvalue
  - And at last we are adding ‘%’ because we are going to pass it as dynamic style, which accepts string type so we used single quotes and perevcentage is used because we are giving height style as % in our css

- Now we are using dynamic styling
  - src > components > Charts > ChartBar.js > ChartBar

```
3   import './ChartBar.css';
4
5   const ChartBar=(props)=>{
6       let barFillHeight='0%';
7
8       if (props.maxValue>0){
9           barFillHeight=Math.round((props.value/props.maxValue)*100)+'%' //here adding '%' because
10      }
11
12      return(
13          <div className='chart-bar'>
14              <div className='chart-bar__capsule'>
15                  <div className='chart-bar__fill' style={{height:barFillHeight}}></div>
16              </div>
17              <div className='chart-bar__label'>{props.label}</div>
18
19          </div>
20      )
21  }
22
23  export default ChartBar;
```