**React.js**

**Table of Contents**:

Introduction: .................................................................................................1

1. What is React?............................................................................1
2. Why we use React?..........................................................2
3. React.js Alternatives..........................................................2
4. React Installation....................................................................... 2
5. Building Single-page-Applications (SPA).................................. 3
6. How is a Component built.........................................................4
7. What is JSX? ......................................................................... 4

6. JSX Limitations......................................................................... 4

7.What are Props? ……...................................................................4

8. Creating my first project as developing a webpage that shows: -   …………………………………………………………………5

9. Outputs ……………………………………………………………….16

**React: -**

**Introduction:**

* React, sometimes referred to as a frontend JavaScript framework, is a JavaScript library created by Facebook. React is a tool for building UI components.
* React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

1. What is React?

* React is a JavaScript Library used for building user interfaces.
* HTML, CSS & JavaScript are about building user interfaces as well.
* React.js is a client-side JavaScript library.
* React allows you to create re-usable and reactive components consisting of HTML and JavaScript (and CSS).
* React.js is all about “Components” (Because all user interfaces in the end are made up of components).
* Components in React are used for reusability, Separation of Concerns.
* React uses Declarative Approach to create Components.

1. Why we use React?

* The main objective of React.js is to develop user Interfaces that improves the speed of the apps.
* It uses Virtual DOM (JavaScript Object) that increases the performance of the object.
* The JavaScript DOM is much faster than Regular Dom.

b.React.js Alternatives:

* React.js is a lean and focused component-based UI library. That’s why certain features are added via community packages.
* Angular: Complete component-based Ui framework, packed with features. Uses TypeScript. Can be an overkill for smaller projects.
* Value.Js: It is a mixture of Angular and React.js. Complete component-based UI framework, includes most core features. A bit less popular than React and Angular.

2. React Installation:

React Environment Setup In this section, we will learn how to set up an environment for the successful development of ReactJS applications.

**ReactJS - Environment Setu**p: -

first, we install: -

Node.Js

VS Code

Npm

**Install Visual Studio Code**: -

Download and install Visual Studio Code from the following URL <https://code.visualstudio.com/download>

The output we have seen when the Project is executed comes from a file called Index.html which resides inside the public folder.

In index.html we have one div tag with id as root.

<div id=”root”></div>

Node.js actually provides a runtime environment to execute JavaScript code from outside a browser. NPM, the default package manager for Nodejs is used for managing and sharing the packages for any JavaScript project. React uses Node.js and NPM for the management of dependencies and runtime

[https://nodejs.org](https://nodejs.org/)

After the installation, check the versions using the below commands.

node -v

npm –v

3. Building Single-Page-Applications: -

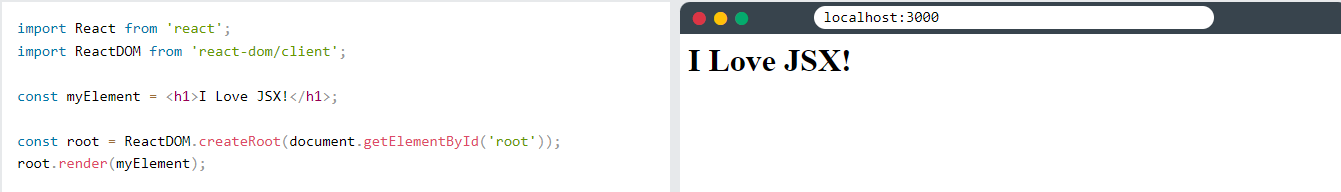
* React can be used to control parts of HTML pages or entire pages.
* React can also be used to control the entire frontend of a web application (Single-Page-Applications).

4.How is a Component Built?

* By Combining HTML, CSS & JavaScript we can build Components.
* Here, CSS is the least important one.

5. What is JSX?

* JSX stands for JavaScript XML.
* JSX is like HTML for JavaScript.
* JSX makes it easier to write and add HTML to React.
* JSX converts HTML tags into React elements.
* It is not important to use JSX but, JSX makes it easier to write React Applications.

**** 

* As, you can see in this example, JSX allows you to write HTML directly within the JavaScript.

6. JSX Limitations:

* You can't return more than one “root” JSX element (you also can't store more than one “root” JSX element in a variable.
* If you wrap the elements with <div>, then you will have only one return value (Doesn’t have to be <div> - Any element will do the trick).

7. What are Props?

* Props are Arguments passed into React Components.
* Props stands for “Properties”.
* Props are passed into Components via HTML attributes.
* Props are also, how you pass the data from one component to another, as parameters.
* Props are Read-only. You will get errors if you try to change their value.

8. Creating my first project as developing a webpage that shows: -

* To Start with my first React.js project, I first downloaded node.js from my internet browser.
* Then I installed it in my PC and went into Command Prompt or Terminal.
* I then entered the following syntax to install npm directories:

**npx create-react-app my-app**

* Instead of my-app, I used my-main-app for the new project.
* Then I opened Visual Studio Code and Opened the My-First-App folder and then opened terminal in Visual studio code.
* Then I entered the following syntax to get work of npm:
* **Npm start**

* This above command starts the app in development mode.

**Folder Structure:**

**Src:**

Components: -

App.js

Expenses.js

Expenses.css

ExpensesDate.js

ExpensesDate.css

ExpensesItem.js

ExpensesItem.css

Chart: -

Chart.js

Chart.css

ChartBar.js

ChatBar.css

New Expense: -

ExpenseChart.js

ExpenseChart.css

ExpenseList.js

ExpenseList.css

ExpenseFilter.js

ExpenseFilter.css

ExpenseForm.js

ExpenseForm.css

NewExpense.js

NewExpense.css

**App.js: -**

// import ExpenseItem from './components/ExpenseItem';

import React, { useState } from "react";

import NewExpense from "./components/NewExpense/NewExpense";

import Expenses from "./components/Expenses";

const DUMMY\_EXPENSES = [

  {

    id: 'e1',

    title: 'Toilet Paper',

    amount: 94.12,

    date: new Date(2020, 7, 14),

  },

  { id: 'e2', title: 'New TV', amount: 799.49, date: new Date(2021, 2, 12) },

  {

    id: 'e3',

    title: 'Car Insurance',

    amount: 294.67,

    date: new Date(2021, 2, 28),

  },

  {

    id: 'e4',

    title: 'New Desk (Wooden)',

    amount: 450,

    date: new Date(2021, 5, 12),

  },

];

const App = () => {

  const [expenses, setExpenses] = useState(DUMMY\_EXPENSES);

  const addExpenseHandler = (expense) => {

    setExpenses((prevExpenses) => {

      return [expense, ...prevExpenses];

    });

  };

  // return React.createElement(

  //   'div',

  //   {},

  //   React.createElement('h2', {}, "Let's get started!"),

  //   React.createElement(Expenses, { items: expenses })

  // );

  return (

    <div>

      <NewExpense onAddExpense={addExpenseHandler} />

      <Expenses items={expenses} />

    </div>

  );

};

export default App;

**Expenses.js: -**

import React, { useState } from 'react';

// import ExpenseItem from "./ExpenseItem";

import Card from "./Card";

import ExpensesFilter from './NewExpense/ExpensesFilter';

import ExpensesList from './NewExpense/ExpensesList';

import './Expenses.css';

import ExpensesChart from './NewExpense/ExpenseChart';

const Expenses = (props) => {

    const [filteredYear, setFilteredYear] = useState('2020');

    const filterChangeHandler = (selectedYear) => {

      setFilteredYear(selectedYear);

    };

    const filteredExpenses = props.items.filter(expense => {

        return expense.date.getFullYear().toString() == filteredYear;

    });

    return (

      <div>

        <Card className='expenses'>

          <ExpensesFilter

            selected={filteredYear}

            onChangeFilter={filterChangeHandler}

          />

          <ExpensesChart expenses={filteredExpenses}/>

          <ExpensesList items={filteredExpenses} />

        </Card>

      </div>

    );

  };

  export default Expenses;

**Expenses.css: -**

.expenses {

    padding: 1rem;

    background-color: rgb(31,31,31);

    margin: 2rem auto;

    width: 50rem;

    max-width: 95%;

}

**ExpensesDate.js: -**

import React from 'react';

import './ExpenseDate.css';

const ExpenseDate = (props) => {

    const month = props.date.toLocaleString('en-US', { month: 'long' });

    const day = props.date.toLocaleString('en-US', { day: '2-digit' });

    const year = props.date.getFullYear();

    return (

      <div className='expense-date'>

        <div className='expense-date\_\_month'>{month}</div>

        <div className='expense-date\_\_year'>{year}</div>

        <div className='expense-date\_\_day'>{day}</div>

      </div>

    );

  };

  export default ExpenseDate;

**ExpensesDate.css: -**

.expense-date {

    display: flex;

    flex-direction: column;

    width: 5.5rem;

    height: 5.5rem;

    border: 1px solid #ececec;

    background-color: #2a2a2a;

    color: white;

    border-radius: 12px;

    align-items: center;

    justify-content: center;

}

.expense-date\_\_month {

    font-size: 0.75rem;

    font-weight: bold;

}

.expense-date\_\_year {

    font-size: 0.75rem;

}

.expense-date\_\_day {

    font-size: 1.5rem;

    font-weight: bold;

}

**ExpensesItem.js: -**

import React, { useState } from 'react';

import './ExpenseItem.css';

import ExpenseDate from './ExpenseDate';

import Card from './Card';

const ExpenseItem = (props) => {

    return (

      <li>

      <Card className='expense-item'>

        <ExpenseDate date={props.date} />

        <div className='expense-item\_\_description'>

          <h2>{props.title}</h2>

          <div className='expense-item\_\_price'>${props.amount}</div>

        </div>

      </Card>

      </li>

    );

  }

  export default ExpenseItem;

**ExpensesItem.css: -**

.expense-item {

    display: flex;

    justify-content: space-between;

    align-items: center;

    /\* box-shadow: 0 2px 8px rgba(0,0,0,0.25); \*/

    padding: 0.5rem;

    margin: 1rem 0;

    /\* border-radius: 12px; \*/

    background-color: #4b4b4b;

}

.expense-item\_\_description {

    display: flex;

    flex-direction: column;

    gap: 1rem;

    align-items: flex-end;

    flex-flow: column-reverse;

    justify-content: flex-start;

    flex: 1;

}

.expense-item h2 {

    color: #3a3a3a;

    font-size: 1rem;

    flex: 1;

    margin: 0 1rem;

    color: white;

}

 .expense-item\_\_price{

    font-size: 1rem;

    font-weight: bold;

    color: white;

    background-color: #40005d;

    border: 1px solid white;

    padding: 0.5rem;

    border-radius: 12px;

 }

@media (min-width:580px) {

    .expense-item\_\_description {

        flex-direction: row;

        align-items: center;

        justify-content: flex-start;

        flex: 1;

    }

    .expense-item\_\_description h2 {

        font-size: 1.25rem;

    }

    .expense-item\_\_price {

        font-size: 1.25rem;

        padding: 0.5rem 1.5rem;

    }

}

**Chart.js: -**

import React from "react";

import './Chart.css';

import ChartBar from "./ChartBar";

const Chart = (props) => {

    const dataPointValues = props.dataPoints.map(dataPoint => dataPoint.value);

    const totalMaximum = Math.max(...dataPointValues);

    return (

      <div className='chart'>

        {props.dataPoints.map((dataPoint) => (

          <ChartBar

            key={dataPoint.label}

            value={dataPoint.value}

            maxValue={totalMaximum}

            label={dataPoint.label}

          />

        ))}

      </div>

    );

  };

  export default Chart;

**Chart.css: -**

.chart {

    padding: 1rem;

    border-radius: 12px;

    background-color: #f8dfff;

    text-align: center;

    display: flex;

    justify-content: space-around;

    height: 10rem;

  }

**ChartBar.js: -**

import React from "react";

import './ChartBar.css';

const ChartBar = (props) => {

    let barFillHeight = '0%';

    if (props.maxValue > 0) {

      barFillHeight = Math.round((props.value / props.maxValue) \* 100) + '%';

    }

    return (

      <div className='chart-bar'>

        <div className='chart-bar\_\_inner'>

          <div

            className='chart-bar\_\_fill'

            style={{ height: barFillHeight }}

          ></div>

        </div>

        <div className='chart-bar\_\_label'>{props.label}</div>

      </div>

    );

  };

  export default ChartBar;

**ChartBar.css: -**

.chart-bar {

    height: 100%;

    display: flex;

    flex-direction: column;

    align-items: center;

  }

  .chart-bar\_\_inner {

    height: 100%;

    width: 100%;

    border: 1px solid #313131;

    border-radius: 12px;

    background-color: #c3b4f3;

    overflow: hidden;

    display: flex;

    flex-direction: column;

    justify-content: flex-end;

  }

  .chart-bar\_\_fill {

    background-color: #4826b9;

    width: 100%;

    transition: all 0.3s ease-out;

  }

  .chart-bar\_\_label {

    font-weight: bold;

    font-size: 0.5rem;

    text-align: center;

  }

**ExpenseChart.js: -**

import React from "react";

import Chart from "../Chart/Chart";

const ExpensesChart = (props) => {

    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 },

    ];

    for (const expense of props.expenses) {

      const expenseMonth = expense.date.getMonth(); // starting at 0 => January => 0

      chartDataPoints[expenseMonth].value += expense.amount;

    }

    return <Chart dataPoints={chartDataPoints} />;

  };

  export default ExpensesChart;

**ExpenseList.js: -**

import React from "react";

import './ExpensesList.css'

import ExpenseItem from "../ExpenseItem";

const ExpensesList = (props) => {

    if (props.items.length === 0) {

      return <h2 className='expenses-list\_\_fallback'>Found no expenses.</h2>;

    }

    return (

      <ul className='expenses-list'>

        {props.items.map((expense) => (

          <ExpenseItem

            key={expense.id}

            title={expense.title}

            amount={expense.amount}

            date={expense.date}

          />

        ))}

      </ul>

    );

  };

  export default ExpensesList;

**ExpenseList.css: -**

.expenses-list {

    list-style: none;

    padding: 0;

  }

  .expenses-list\_\_fallback {

    color: white;

    text-align: center;

  }

**ExpenseFilter.js: -**

import React from "react";

import './ExpensesFilter.css';

const ExpensesFilter = (props) => {

    const dropdownChangeHandler = (event) => {

      props.onChangeFilter(event.target.value);

    };

    return (

      <div className='expenses-filter'>

        <div className='expenses-filter\_\_control'>

          <label>Filter by year</label>

          <select value={props.selected} onChange={dropdownChangeHandler}>

            <option value='2022'>2022</option>

            <option value='2021'>2021</option>

            <option value='2020'>2020</option>

            <option value='2019'>2019</option>

          </select>

        </div>

      </div>

    );

  };

  export default ExpensesFilter;

**ExpenseFilter.css: -**

.expenses-filter {

    color: white;

    padding: 0 1rem;

  }

  .expenses-filter\_\_control {

    display: flex;

    width: 100%;

    align-items: center;

    justify-content: space-between;

    margin: 1rem 0;

  }

  .expenses-filter label {

    font-weight: bold;

    margin-bottom: 0.5rem;

  }

  .expenses-filter select {

    font: inherit;

    padding: 0.5rem 3rem;

    font-weight: bold;

    border-radius: 6px;

  }

**ExpenseForm.js: -**

import React, { useState } from "react";

import './ExpenseForm.css';

function ExpenseForm(props) {

    // document.getElementById('').addEventListener('click', (event) => {})

    const [enteredTitle, setEnteredTitle] = useState('');

    const [enteredAmount, setEnteredAmount] = useState('');

    const [enteredDate, setEnteredDate] = useState('');

    // const [userInput, setUserInput] = useState({

    //     enteredTitle: '',

    //     enteredAmount: '',

    //     enteredDate: ''

    // });

    const titleChangeHandler = (event) => {

        setEnteredTitle(event.target.value);

        // setUserInput({

        //     ...userInput,

        //     enteredTitle: event.target.value,

        // });

        // setUserInput((prevState) => {

        //     return {...prevState, enteredTitle: event.target.value};

        // });

    };

    const amountChangeHandler = (event) => {

        setEnteredAmount(event.target.value);

        // setUserInput({

        //     ...userInput,

        //     enteredAmount: event.target.value,

        // });

        // setUserInput((prevState) => {

        //     return {...prevState, enteredAmount: event.target.value};

        // });

    };

    const dateChangeHandler = (event) => {

        setEnteredDate(event.target.value);

        // setUserInput({

        //     ...userInput,

        //     enteredDate: event.target.value,

        // });

        // setUserInput((prevState) => {

        //     return {...prevState, enteredDate: event.target.value};

        // });

    };

        const submitHandler = (event) => {

            event.preventDefault();

            const expenseData = {

                title: enteredTitle,

                amount: enteredAmount,

                date: new Date(enteredDate)

            };

            // console.log(ExpenseData);

            props.onSaveExpenseData(expenseData);

            setEnteredTitle('');

            setEnteredAmount('');

            setEnteredDate('');

        };

    return (

    <form onSubmit={submitHandler}>

        <div className="new-expense\_\_controls">

            <div className="new-expense\_\_control">

                <label>Title</label>

                <input

                type='text'

                value={enteredTitle}

                onChange={titleChangeHandler}

                />

            </div>

            <div className="new-expense\_\_control">

                <label>Amount</label>

                <input type='number'

                 min="0.01"

                 step="0.01"

                 value={enteredAmount}

                  onChange={amountChangeHandler}

                   />

            </div>

            <div className="new-expense\_\_control">

                <label>Date</label>

                <input type='date'

                 min="2019-01-01"

                 max="2022-12-31"

                 value={enteredDate}

                  onChange={dateChangeHandler}

                   />

            </div>

        </div>

        <div className="new-expense\_\_actions">

            <button type="button" onClick={props.onCancel}>Cancel</button>

            <button type='submit'>Add Expense</button>

        </div>

    </form>

)};

export default ExpenseForm;

**ExpenseForm.css: -**

.new-expense\_\_controls {

    display: flex;

    flex-wrap: wrap;

    gap: 1rem;

    margin-bottom: 1rem;

    text-align: left;

}

.new-expense\_\_control label {

   font-weight: bold;

   margin-bottom: 0.5rem;

   display: block;

}

.new-expense\_\_control input {

  font: inherit;

  padding: 0.5rem;

  border-radius: 6px;

  border: 1px solid #ccc;

  width: 20rem;

  max-width: 100%;

}

.new-expense\_\_actions {

    text-align: right;

}

**NewExpense.js: -**

import React, { useState } from "react";

import ExpenseDate from "../ExpenseDate";

import ExpenseForm from "./ExpenseForm";

import './NewExpense.css';

const NewExpense = (props) => {

    const [isEditing, setIsEditing] = useState(false);

    const saveExpenseDataHandler = (enteredExpenseData) => {

      const expenseData = {

        ...enteredExpenseData,

        id: Math.random().toString(),

      };

      props.onAddExpense(expenseData);

      setIsEditing(false);

    };

    const startEditingHandler = () => {

      setIsEditing(true);

    };

    const stopEditingHandler = () => {

      setIsEditing(false);

    };

    return (

      <div className='new-expense'>

        {!isEditing && (

          <button onClick={startEditingHandler}>Add New Expense</button>

        )}

        {isEditing && (

          <ExpenseForm

            onSaveExpenseData={saveExpenseDataHandler}

            onCancel={stopEditingHandler}

          />

        )}

      </div>

    );

  };

  export default NewExpense;

**NewExpense.css: -**

.new-expense {

    background-color: #a892ee;

    padding: 1rem;

    margin: 2rem auto;

    width: 50rem;

    max-width: 95%;

    border-radius: 12px;

    text-align: center;

    box-shadow: 0 1px 8px rgba(0, 0, 0, 0.25);

}

.new-expense button {

    font: inherit;

    cursor: pointer;

    padding: 1rem 2rem ;

    border: 1px solid #40005d;

    background-color: #40005d;

    color: white;

    border-radius: 12px;

    margin-right: 1rem;

}

.new-expense button:hover,

.new-expense button:active {

    background-color: #510674;

    border-color: #510674;

}

.new-expense button.alternative {

    color: #220131;

    border-color: transparent;

    background-color: transparent;

}

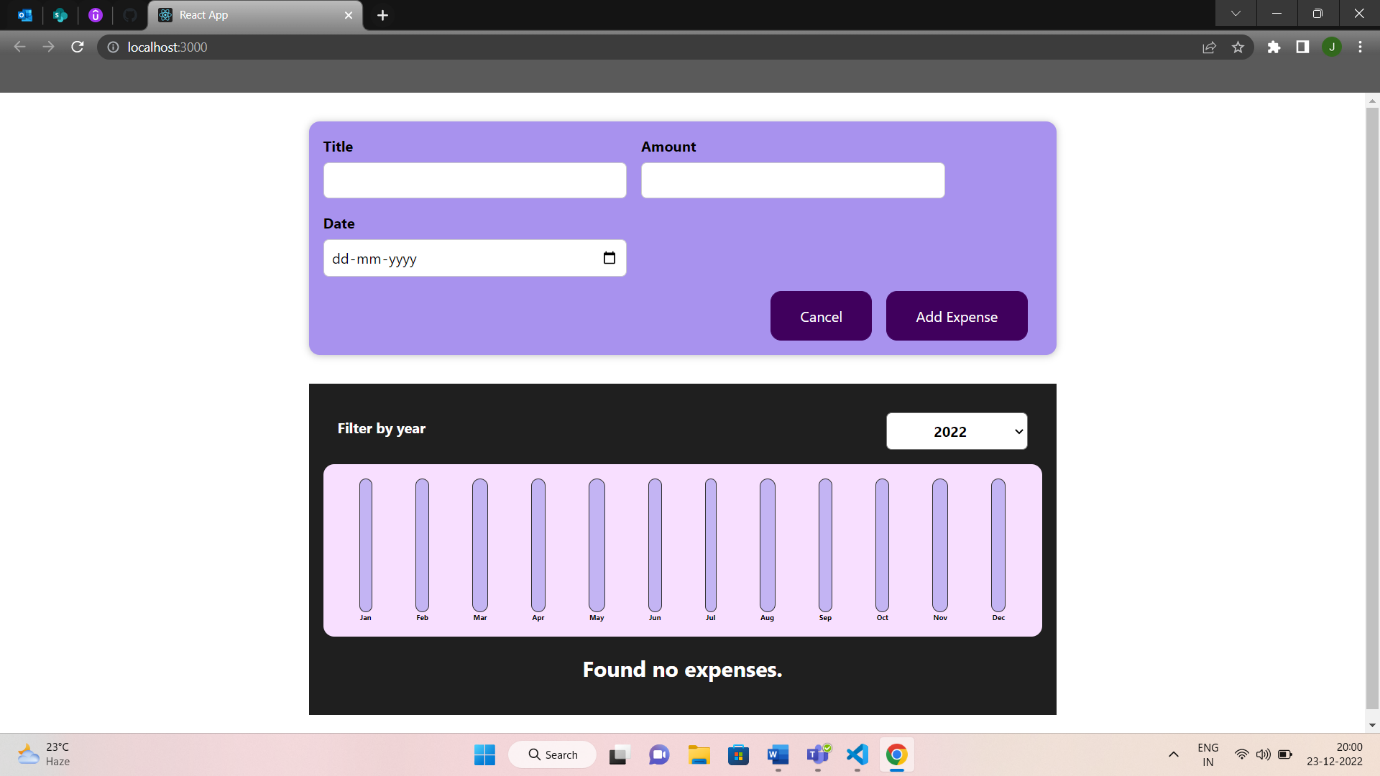
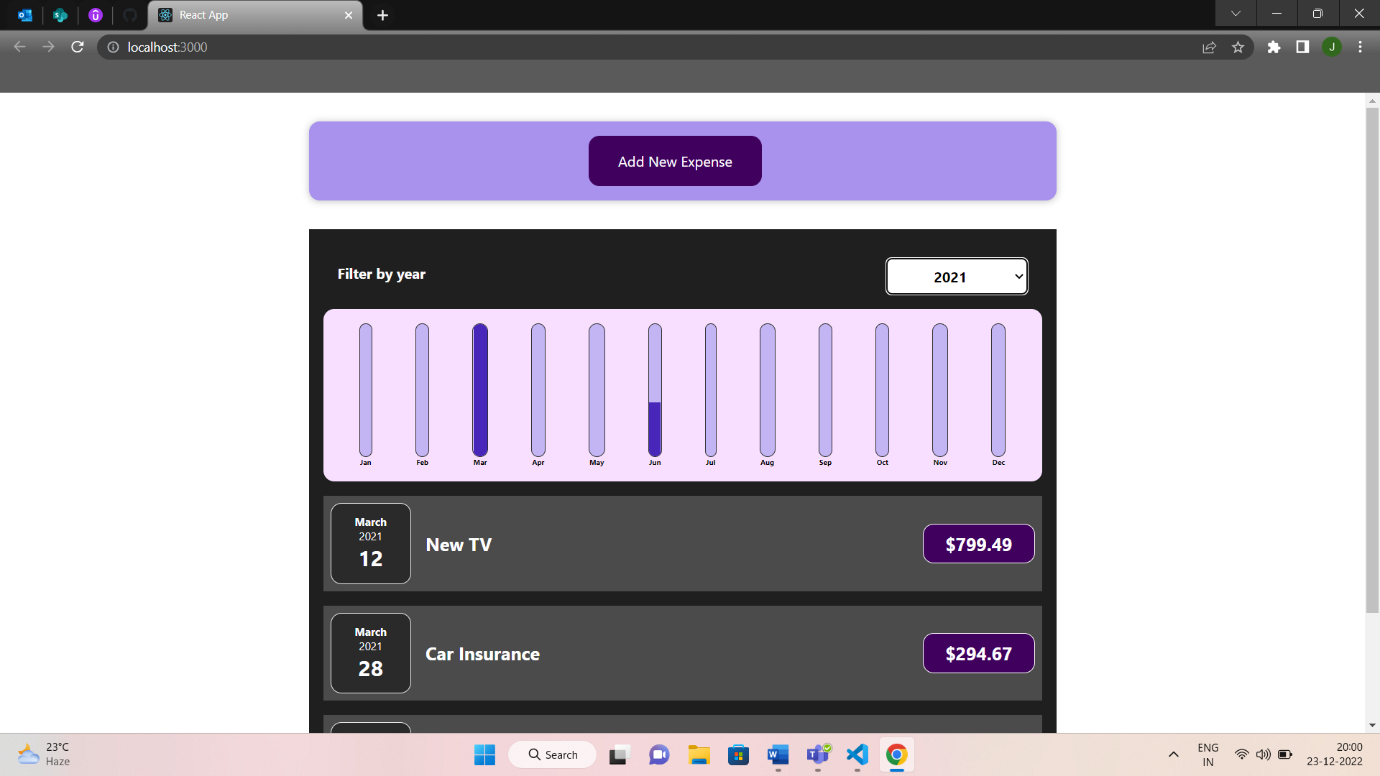
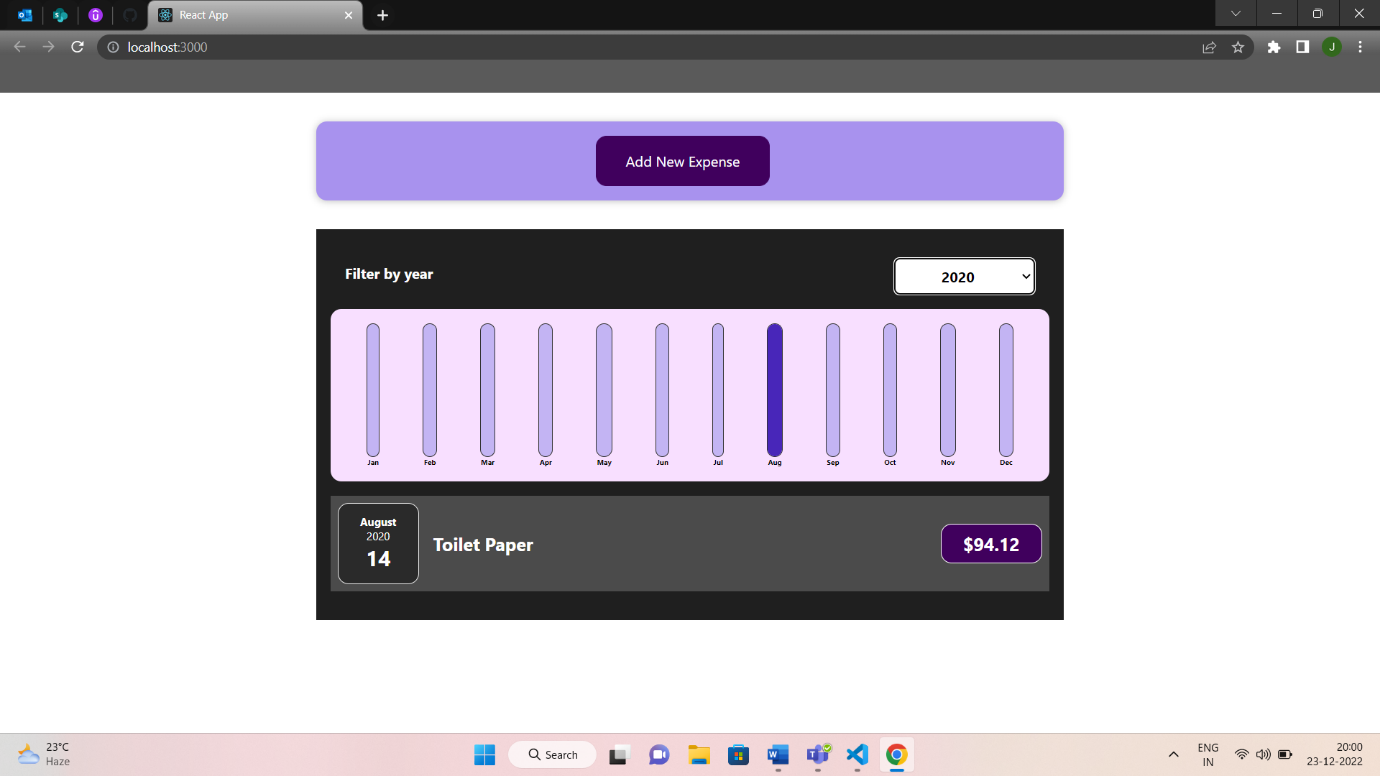
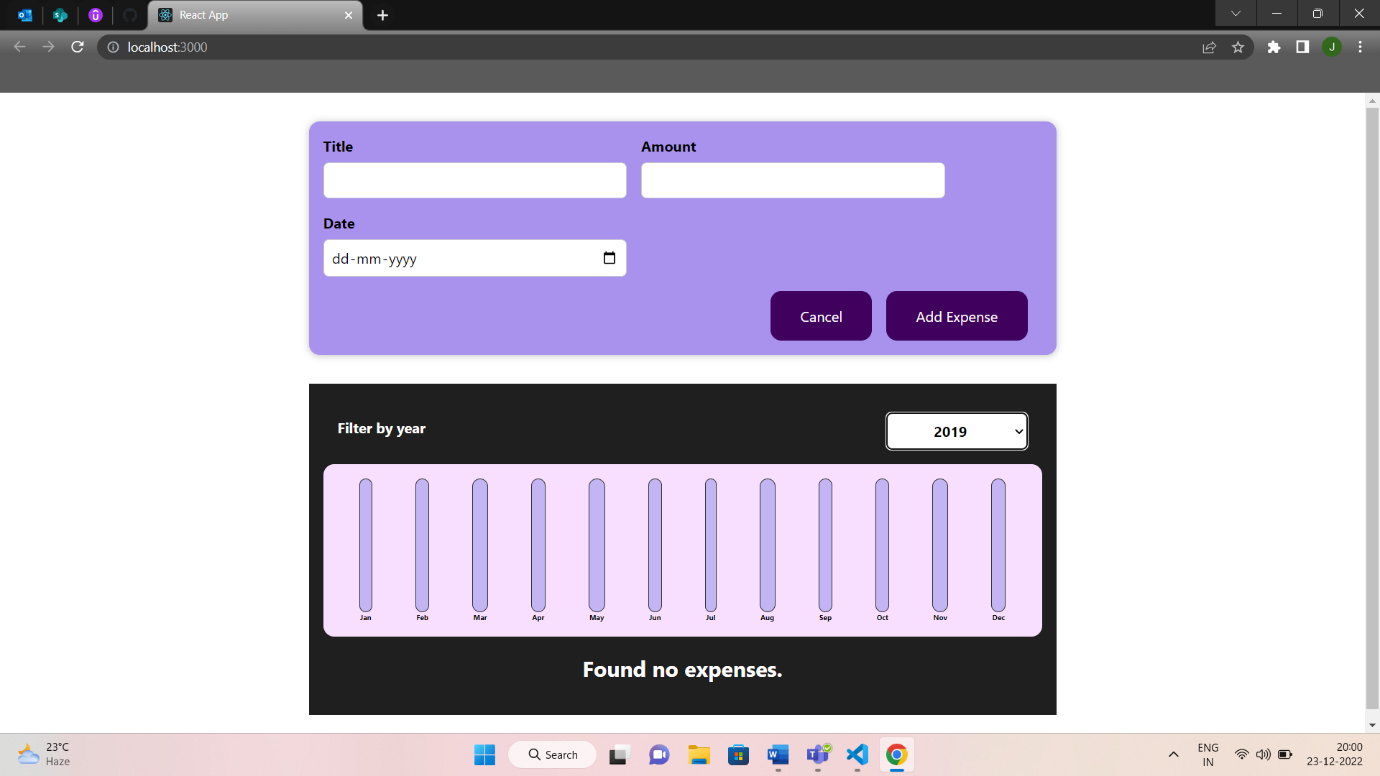
.new-expense button.alternative:hover,

.new-expense button.alternative:active {

    background-color: #ddb3f8;

}

**9. Outputs: -**

****

***Thank You***