# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ”ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 6

з дисципліни

« Екстримальне програмування»

Варіант 25

Студент групи КН-311

Прокіпчук О.А.

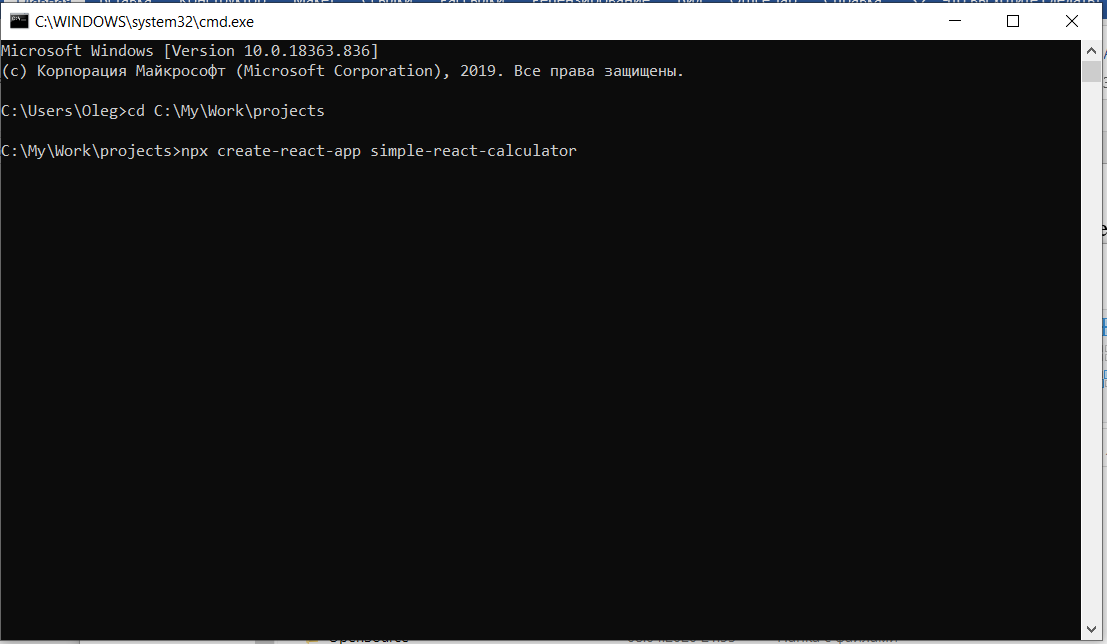
Прийняв викладач

Щербак С. С.

Львів-2020

**Мета роботи:** розробити калькулятор на базі Nodejs та ReactJs, впевнитись в правильності його обчислень.

**Хід роботи**



Для виконання усіх вимог створюю 3 компоненти:

* CalculatorPage
* Calculator
* ResultTable

**export class** Calculator **extends** React.Component{  
  
 constructor(props) {  
 **super**(props);  
  
 **this**.**state** = {  
  
 };  
 }  
  
 componentDidMount() {  
 **let** operand1 = ***document***.getElementById(**"operand1"**);  
 **let** operand2 = ***document***.getElementById(**"operand2"**);  
 **let** operator = ***document***.getElementById(**"operator"**);  
  
 **this**.*setInputFilter*(operand1, **this**.*floatInputFilter*);  
 **this**.*setInputFilter*(operand2, **this**.*floatInputFilter*);  
 **this**.*setInputFilter*(operator, **this**.*operatorInputFilter*);  
 }  
  
 *isPrime* = (num) => {  
 **for**(**let** i = 2; i < num; i++)  
 **if**(num % i === 0) **return false**;  
 **return** num > 1;  
 };  
  
 *getHighestPrimeBetweenTwoNumbers* = (num1, num2) => {  
 **if** (num2 < 0) **return 'err'**;  
  
 **let** numm1 = ***Math***.floor(num1);  
 **let** numm2 = ***Math***.floor(num2);  
  
 **for** (**let** i = numm2; i > numm1; i--) {  
 **if** (i < 0) **return 'err'**;  
 **if** (**this**.*isPrime*(i)) **return** i;  
 }  
  
 **return 'err'**;  
 };  
  
 *floatInputFilter* = (value) => {  
 **if** (value === **''** || value === **'-'** || value === **'+'**) **return true**;  
 **return** /^[+-]?\d+(\.)?(\d+)?$/.test(value);  
 };  
  
 *operatorInputFilter* = (value) => {  
 **return** /^[+/%@\*-^]?$/.test(value);  
 };  
  
 *setInputFilter* = (textbox, inputFilter) => {  
 [**"input"**, **"keydown"**, **"keyup"**, **"mousedown"**, **"mouseup"**, **"select"**, **"contextmenu"**, **"drop"**].forEach(**function**(event) {  
 textbox.addEventListener(event, **function**(event) {  
 ***console***.log(**this**.**value**);  
 **if** (inputFilter(**this**.**value**)) {  
 **this**.**oldValue** = **this**.**value**;  
 **this**.**oldSelectionStart** = **this**.**selectionStart**;  
 **this**.**oldSelectionEnd** = **this**.**selectionEnd**;  
 } **else if** (**this**.hasOwnProperty(**"oldValue"**)) {  
 **this**.**value** = **this**.**oldValue**;  
 **this**.setSelectionRange(**this**.**oldSelectionStart**, **this**.**oldSelectionEnd**);  
 } **else** {  
 **this**.**value** = **""**;  
 }  
 });  
 });  
 };  
  
 *onCalculateClick* = (event) => {  
 **let** calculateForm = event.**target**.**parentElement**.**parentElement**;  
   
 **let** operand1 = +calculateForm.operand1.**value**;  
 **let** operand2 = +calculateForm.operand2.**value**;  
 **let** operator = calculateForm.**operator**.**value**;  
  
 **let** result = **""**;  
  
 **switch** (operator) {  
 **case '+'**:  
 result = operand1 + **" + "** + operand2 + **" = "** + (operand1 + operand2);  
 **break**;  
 **case '/'**:  
 result = operand1 + **" / "** + operand2 + **" = "** + (operand1 / operand2);  
 **break**;  
 **case '%'**:  
 result = operand1 + **" % "** + operand2 + **" = "** + (operand1 % operand2);  
 **break**;  
 **case '\*'**:  
 result = operand1 + **" \* "** + operand2 + **" = "** + (operand1 \* operand2);  
 **break**;  
 **case '-'**:  
 result = operand1 + **" - "** + operand2 + **" = "** + (operand1 - operand2);  
 **break**;  
 **case '^'**:  
 result = operand1 + **" ^ "** + operand2 + **" = "** + (***Math***.pow(operand1, operand2));  
 **break**;  
 **case '@'**:  
 result = operand1 + **" @ "** + operand2 + **" = "** + (**this**.*getHighestPrimeBetweenTwoNumbers*(operand1, operand2));  
 **break**;  
 **default**:  
 **break**;  
 }  
  
 **this**.**props**.*onReceiveResult*(result);  
 };  
  
 render() {  
 **return**(  
 <**div className="calculator-body"**>  
 <**form**>  
 <**div className="calculator-body-input-group"**>  
 <**div className="form-group"**>  
 <**input type="text" id="operand1" name="operand1" className="calculator-operand form-control" defaultValue="0" placeholder="Operand 1"**/>  
 </**div**>  
 <**div className="form-group"**>  
 <**input type="text" id="operator" name="operator" className="calculator-operator form-control"**/>  
 </**div**>  
 <**div className="form-group"**>  
 <**input type="text" id="operand2" name="operand2" className="calculator-operand form-control" defaultValue="0" placeholder="Operand 2"**/>  
 </**div**>  
 </**div**>  
 <**div className="calculator-body-button-container"**>  
 <**button onClick=**{**this**.*onCalculateClick*} **type="button" className="calculate-button btn btn-dark"**>Calculate</**button**>  
 </**div**>  
 </**form**>  
 </**div**>  
 );  
 }  
  
  
}

**export class** ResultsTable **extends** React.Component{  
  
 constructor(props) {  
 **super**(props);  
  
 **this**.**state** = {  
 **results**: [],  
 };  
 }  
  
 *addResult* = (result) => {  
 **return this**.setState((state) => {  
 **return** {  
 **results**: [...state.**results**, result]  
 };  
 });  
 };  
  
 render() {  
 **return**(  
 <**table className="results-table table"**>  
 <**thead className="thead-light"**>  
 <**tr**>  
 <**th scope="col"**>#</**th**>  
 <**th scope="col"**>Result</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {**this**.**state**.**results**.map((item, index) =>  
 <**tr key=**{index}>  
 <**th scope="row"**>{index + 1}</**th**>  
 <**td**>{item}</**td**>  
 </**tr**>  
 )}  
 </**tbody**>  
 </**table**>  
 );  
 }  
  
}

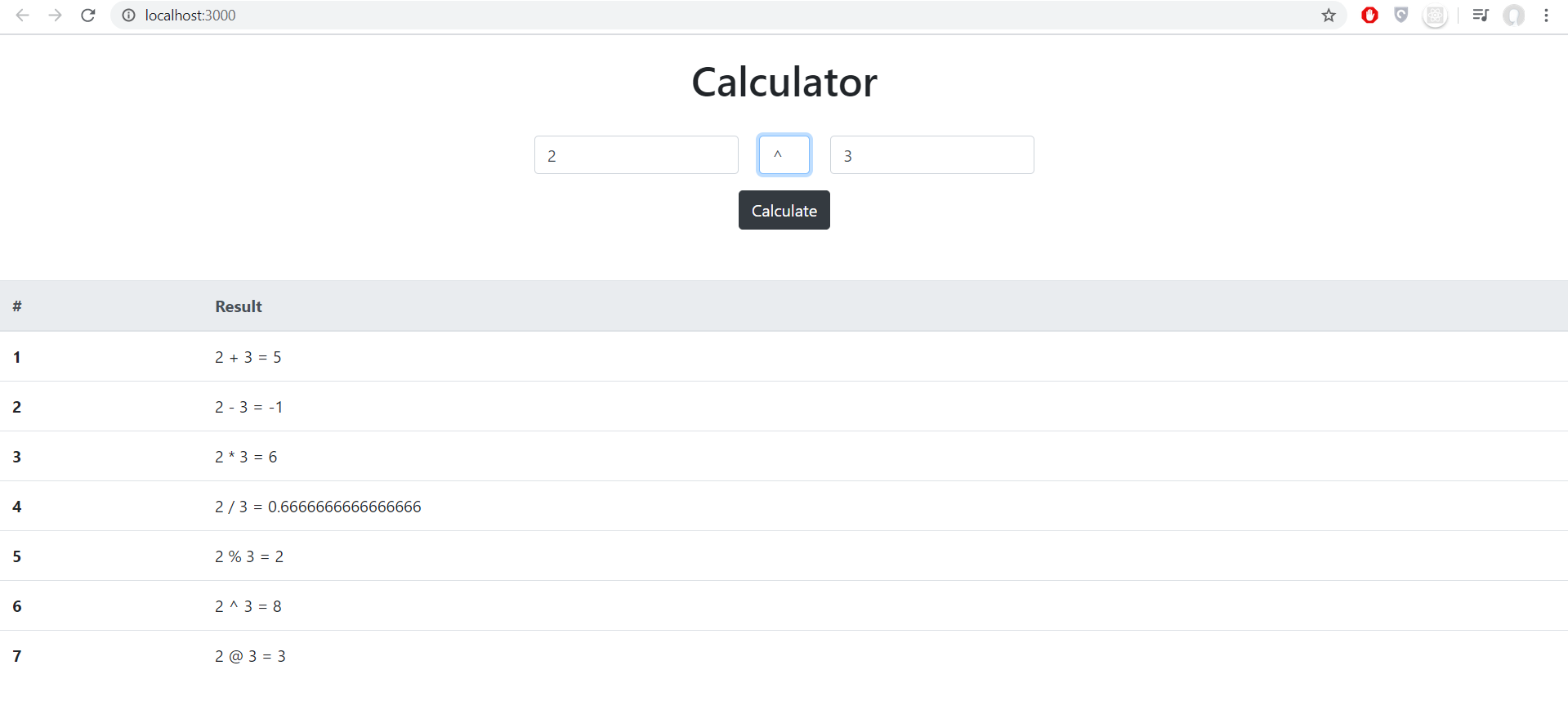
**export class** CalculatorPage **extends** React.Component{  
  
 constructor(props) {  
 **super**(props);  
  
 **this**.**resultsTable** = React.*createRef*();  
 }  
  
 *onReceiveResult* = (result) => {  
 **this**.**resultsTable**.**current**.*addResult*(result);  
 };  
  
 render() {  
 **return**(  
 <**div className="calculator-page"**>  
 <**h1 style=**{{**textAlign**: **"center"**, **marginTop**: **"20px"**}}>Calculator</**h1**>  
 <**Calculator onReceiveResult=**{**this**.*onReceiveResult*}/>  
 <**ResultsTable ref=**{**this**.**resultsTable**}/>  
 </**div**>  
 );  
 }  
  
  
}

Створюю необхідні стилі:

.**results-table** {  
 **margin-top**: 50**px**;  
}

.**calculator-body-input-group** {  
 **margin-top**: 30**px**;  
 **justify-content**: **center**;  
 **display**: **flex**;  
}  
  
.**calculator-body-input-group** > **div** {  
 **display**: **inline-block**;  
}  
  
.**calculator-operand** {  
 **width**: 200**px**;  
}  
  
.**calculator-operator** {  
 **width**: 50**px**;  
 **margin-left**: 20**px**;  
 **margin-right**: 20**px**;  
}  
  
.**calculator-body-button-container** {  
 **justify-content**: **center**;  
 **display**: **flex**;  
}

Тестую додаток:



**Висновок:** на даній лабораторній роботі я розробив калькулятор на базі Nodejs та ReactJs, впевнився в правильності його обчислень.