**1. Write an arrow function that will check if a given number is prime or not. If prime the function will return true and false otherwise. Call the function and display messages accordingly.**

**Ans-**

//21BCSE30 | ANIKET PATRA

//arrow function to check prime or not

const isPrime = (n) => {

if (n === 1 || n === 0) return false;

for (let i = 2; i <= Math.sqrt(n); i++) if (n % i === 0) return false;

return true;

};

let n = 5;

if (isPrime(n)) {

console.log("Prime");

} else {

console.log("Not Prime");

}

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q1.js"

Prime

**2. Write a one-liner arrow function (Lambda) that squares a given number.**

**Ans-**

//21BCSE30 | ANIKET PATRA

const square = (x) => x \* x;

console.log(square(5));

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q2.js"

25

**3. Implement a function calculate that takes two numbers and a callback function. The callback function should perform a mathematical operation (e.g., add, subtract, multiply, divide) on the two numbers.**

**Ans-**

//21BCSE30 | ANIKET PATRA

function calculate(a, b, callback) {

return callback(a, b);

}

function add(a, b) {

return a + b;

}

function subtract(a, b) {

return a - b;

}

function multiply(a, b) {

return a \* b;

}

function divide(a, b) {

if (b !== 0) {

return a / b;

} else {

return undefined;

}

}

console.log(calculate(5, 3, add));

console.log(calculate(5, 3, subtract));

console.log(calculate(5, 3, multiply));

console.log(calculate(6, 3, divide));

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q3.js"

8

2

15

2

**4. Use HTML and JS to display the time (clock) of three different countries.**

**Ans-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>21BCSE30 | ANIKET PATRA</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

}

.clock {

display: inline-block;

margin: 20px;

padding: 20px;

border: 2px solid #ccc;

border-radius: 10px;

}

</style>

</head>

<body>

New York :

<div class="clock" id="clock1"></div>

India :

<div class="clock" id="clock2"></div>

Sydney :

<div class="clock" id="clock3"></div>

<script>

function updateClock(elementId, offset) {

const now = new Date();

document.getElementById(elementId).textContent = now.toLocaleTimeString(

"en-US",

{ timeZone: offset }

);

}

function updateClocks() {

updateClock("clock1", "America/New\_York");

updateClock("clock2", "Asia/Kolkata");

updateClock("clock3", "Australia/Sydney");

}

// Update every second

setInterval(updateClocks, 1000);

// Initial update

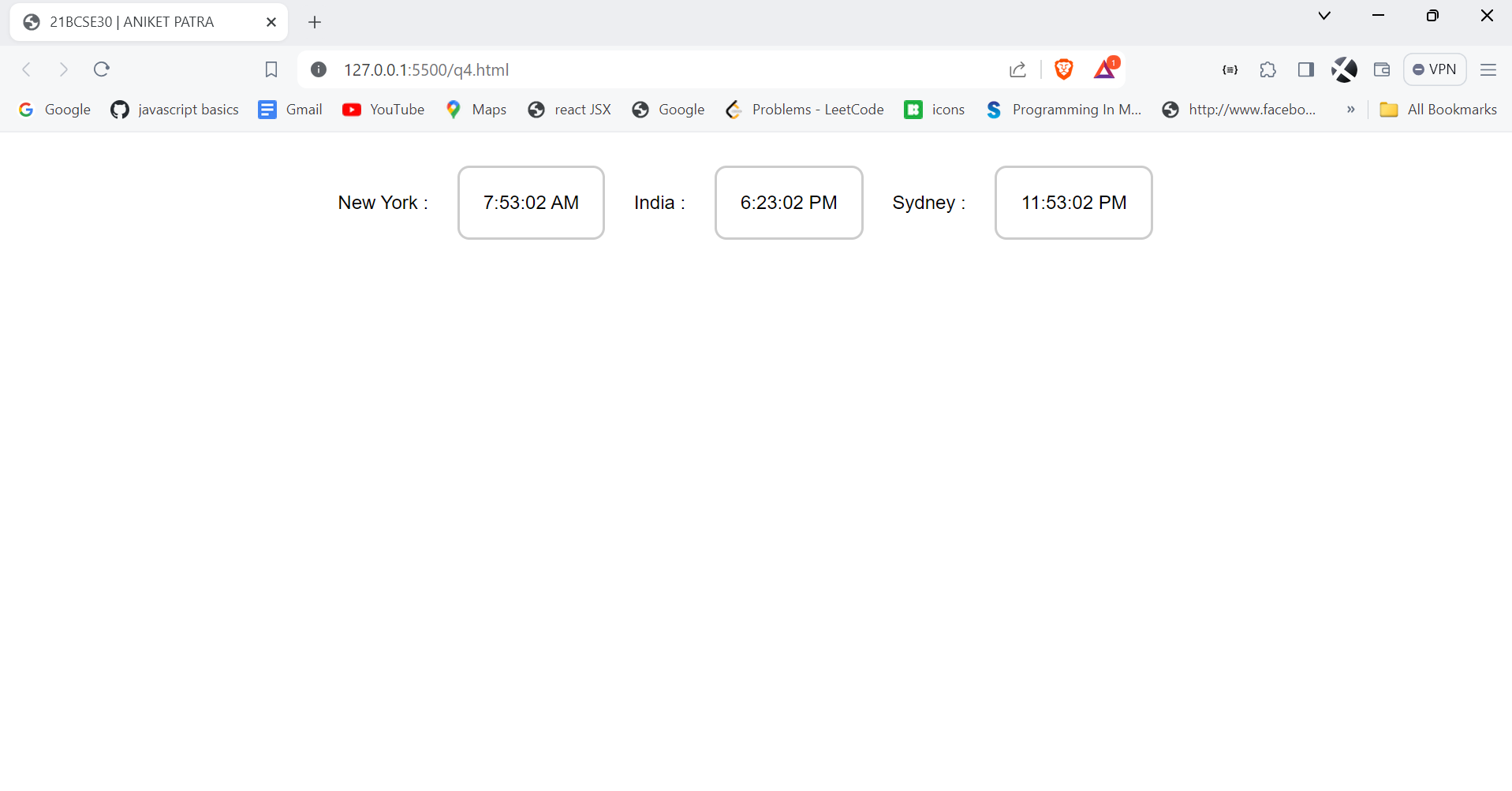
updateClocks();

</script>

</body>

</html>

**Output:-**

****

**5. Consider an array of fruits:**

**fruits = ["apple", "banana", "orange", "grape", "kiwi"]**

**a. Create a function which will take an array and display the array elements. Use this to display the array after each following operation.**

**b. Add one fruit to the fruits array**

**c. Remove the last fruit item from the array.**

**d. Check if the fruits array contains orange.**

**e. Sort the fruits in alphabetical order using sort and localeCompare**

**f. Make shallow copy of the fruits array.**

**Ans-**

//21BCSE30 | ANIKET PATRA

const fruits = ["apple", "banana", "orange", "grape", "kiwi"];

function display(arr) {

// for (let x of arr) {

// console.log(x);

// }

console.log(arr);

}

fruits.push("strawberry");

display(fruits);

fruits.pop();

display(fruits);

if (fruits.includes("orange")) {

console.log("Orange is present in fruits");

} else {

console.log("Orange is not present in fruits");

}

fruits.sort();

display(fruits);

fruits.sort((a, b) => a.localeCompare(b));

display(fruits);

const deepCopy = [...fruits];

console.log(deepCopy);

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q5.js"

[ 'apple', 'banana', 'orange', 'grape', 'kiwi', 'strawberry' ]

[ 'apple', 'banana', 'orange', 'grape', 'kiwi' ]

Orange is present in fruits

[ 'apple', 'banana', 'grape', 'kiwi', 'orange' ]

[ 'apple', 'banana', 'grape', 'kiwi', 'orange' ]

[ 'apple', 'banana', 'grape', 'kiwi', 'orange' ]

**6. Construct an array of 10 numbers and perform the following operations –**

**a. Display the array elements using forEach**

**b. Produce a new array by squaring each number of the given array using map**

**c. Produce an array with all the even numbers present on the original array using filter**

**d. Remove any number from the array using filter**

**e. Sort the array in both ascending and descending order and display using sort**

**Ans-**

//21BCSE30 | ANIKET PATRA

const numbers = [12, 23, 14, 35, 67, 68, 90, 99, 54, 11];

numbers.forEach((num) => {

console.log(num);

});

const squareNumbers = numbers.map((num) => num \* num);

console.log(squareNumbers);

const evenNumbers = numbers.filter((num) => num % 2 === 0);

console.log(evenNumbers);

const randomNumberIndex = Math.floor(Math.random() \* numbers.length);

const randomNumber = numbers[randomNumberIndex];

console.log(randomNumber);

const removedArray = numbers.filter((num) => num !== randomNumber);

console.log(removedArray);

numbers.sort();

console.log("The sorted array in ascending order is : ", numbers);

numbers.sort((a, b) => b - a);

console.log("The sorted array in descending order is : ", numbers);

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q6.js"

12

23

14

35

67

68

90

99

54

11

[

144, 529, 196,

1225, 4489, 4624,

8100, 9801, 2916,

121

]

[ 12, 14, 68, 90, 54 ]

23

[

12, 14, 35, 67, 68,

90, 99, 54, 11

]

The sorted array in ascending order is : [

11, 12, 14, 23, 35,

54, 67, 68, 90, 99

]

The sorted array in descending order is : [

99, 90, 68, 67, 54,

35, 23, 14, 12, 11

]

**7. Consider an array of objects given where each object contains the student's name and age. Perform the following operations using higher order array methods.**

**a. Use forEach loop, object de-structuring, and string literals to display the details in the given format.**

**b. Find the student with the highest age**

**c. Find the average age of the class**

|  |  |
| --- | --- |
| **Ex (for a): Input** | **Output** |
| **const students = [**  **{ name: "Amit", age: 23 },**  **{ name: "Sima", age: 21 },**  **{ name: "Bimal", age: 19 },**  **]**  **Ans-** | **Amit is 23 years old**  **Sima is 21 years old**  **Bimal is 19 years old** |

// 21BCSE30 | ANIKET PATRA

const students = [

{

name: "John",

age: "20",

},

{

name: "Emily",

age: "22",

},

{

name: "Michael",

age: "19",

},

{

name: "Jessica",

age: "21",

},

{

name: "Daniel",

age: "20",

},

{

name: "Sarah",

age: "23",

},

{

name: "David",

age: "18",

},

{

name: "Olivia",

age: "20",

},

{

name: "Matthew",

age: "22",

},

{

name: "Sophia",

age: "19",

},

];

//display the student details

students.forEach((student) => {

const { name, age } = student;

console.log(`${name} is ${age} years old.`);

});

//student with highest age

const studentHighestAge = students.reduce(

(studentHighest, student) =>

student.age > studentHighest.age ? student : studentHighest,

students[0]

);

console.log(`Student with Highest Age is ${studentHighestAge.name} and her age is ${studentHighestAge.age} years.`);

//average age of the students

const sumAges = students.reduce((sum, student) => sum + parseInt(student.age), 0);

console.log("The average age of the students is ", sumAges / students.length);

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q7.js"

John is 20 years old.

Emily is 22 years old.

Michael is 19 years old.

Jessica is 21 years old.

Daniel is 20 years old.

Sarah is 23 years old.

David is 18 years old.

Olivia is 20 years old.

Matthew is 22 years old.

Sophia is 19 years old.

Student with Highest Age is Sarah and her age is 23 years.

The average age of the students is 20.4

**8. Take any string and form an array of objects where each object contains the word and its length.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **"Hello, this is a sample string"** | **[**  **{ word: 'Hello,', length: 6 },**  **{ word: 'this', length: 4 },**  **{ word: 'is', length: 2 },**  **{ word: 'a', length: 1 },**  **{ word: 'sample', length: 6 },**  **{ word: 'string', length: 6 }**  **]** |

**Ans-**

//21BCSE30 | ANIKET PATRA

function generateWordLengthArray(inputString) {

const words = inputString.split(" ");

const result = words.map((word) => {

return {

word: word.replace(/[.,]/g, ""),

length: word.replace(/[.,]/g, "").length,

};

});

return result;

}

const inputString = "Hello, this is a sample string";

const output = generateWordLengthArray(inputString);

console.log(output);

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q8.js"

[

{ word: 'Hello', length: 5 },

{ word: 'this', length: 4 },

{ word: 'is', length: 2 },

{ word: 'a', length: 1 },

{ word: 'sample', length: 6 },

{ word: 'string', length: 6 }

]

**9. Consider an array consisting of the Employee objects of the following structure and solve the following problem.**

**{name: “emp name”, skill: [‘skill1’, ‘skill2’, ‘skill3’], salary: 12345}**

**a. Display the name of the employee and the technical skills they have**

**b. Display the employee details in sorted order of their name**

**c. Find the highest-paid employee**

**d. Display all the unique skills available in the company**

**e. Find all the employees who know JavaScript.**

**Ans-**

//21BCSE30 | ANIKET PATRA

const Employees = [

{

name: "Aniket Patra",

skill: ["c", "c++", "java", "javascript", "php"],

salary: 90000,

},

{ name: "Anmol Partha Behera", skill: ["c++", "java"], salary: 70000 },

{ name: "Ashutosh Panda", skill: ["python", "php", "java"], salary: 60000 },

{ name: "Ram Rastogi", skill: ["c#", "asp.net"], salary: 85000 },

{ name: "Abhinash Sharma", skill: ["c#", "sql"], salary: 120000 },

{ name: "Priyanshu Rout", skill: ["java", "python"], salary: 100000 },

{

name: "Kalyani Nayak",

skill: ["sql", "javascript", "java", "python"],

salary: 150000,

},

{ name: "Kasturi Dhal", skill: ["php", "html/css"], salary: 95000 },

{

name: "Arya Dash",

skill: ["c", "c++", "java", "javascript"],

salary: 200000,

},

{ name: "Anubhav Mohanty", skill: ["python", "sql"], salary: 50000 },

];

Employees.forEach((employee) => {

const { name, skill } = employee;

console.log(`Employee Name : ${name}`);

skill.forEach((skillName) => console.log(`Skills : ${skillName}`));

});

const sortedEmployees = Employees.sort((a, b) => a.name.localeCompare(b.name));

console.log(sortedEmployees);

const HighestPaidEmployee = Employees.reduce(

(hpaid, employee) => (hpaid.salary > employee.salary ? hpaid : employee),

Employees[0]

);

console.log(

`The Highest Paid Employee is ${HighestPaidEmployee.name} with a salary of ${HighestPaidEmployee.salary}`

);

var skills = new Set();

Employees.forEach((employee) => {

employee.skill.forEach((skill) => skills.add(skill));

});

console.log("Unique Skills among Employees are ", [...skills]);

const peopleKnowJavascript = Employees.filter((employee) => employee.skill.includes("javascript"));

console.log('People who know javascript are : ');

peopleKnowJavascript.forEach((employee) => {

console.log(employee.name);

});

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q9.js"

Employee Name : Aniket Patra

Skills : c

Skills : c++

Skills : java

Skills : javascript

Skills : php

Employee Name : Anmol Partha Behera

Skills : c++

Skills : java

Employee Name : Ashutosh Panda

Skills : python

Skills : php

Skills : java

Employee Name : Ram Rastogi

Skills : c#

Skills : asp.net

Employee Name : Abhinash Sharma

Skills : c#

Skills : sql

Employee Name : Priyanshu Rout

Skills : java

Skills : python

Employee Name : Kalyani Nayak

Skills : sql

Skills : javascript

Skills : java

Skills : python

Employee Name : Kasturi Dhal

Skills : php

Skills : html/css

Employee Name : Arya Dash

Skills : c

Skills : c++

Skills : java

Skills : javascript

Employee Name : Anubhav Mohanty

Skills : python

Skills : sql

[

{ name: 'Abhinash Sharma', skill: [ 'c#', 'sql' ], salary: 120000 },

{

name: 'Aniket Patra',

skill: [ 'c', 'c++', 'java', 'javascript', 'php' ],

salary: 90000

},

{

name: 'Anmol Partha Behera',

skill: [ 'c++', 'java' ],

salary: 70000

},

{

name: 'Anubhav Mohanty',

skill: [ 'python', 'sql' ],

salary: 50000

},

{

name: 'Arya Dash',

skill: [ 'c', 'c++', 'java', 'javascript' ],

salary: 200000

},

{

name: 'Ashutosh Panda',

skill: [ 'python', 'php', 'java' ],

salary: 60000

},

{

name: 'Kalyani Nayak',

skill: [ 'sql', 'javascript', 'java', 'python' ],

salary: 150000

},

{ name: 'Kasturi Dhal', skill: [ 'php', 'html/css' ], salary: 95000 },

{

name: 'Priyanshu Rout',

skill: [ 'java', 'python' ],

salary: 100000

},

{ name: 'Ram Rastogi', skill: [ 'c#', 'asp.net' ], salary: 85000 }

]

The Highest Paid Employee is Arya Dash with a salary of 200000

Unique Skills among Employees are [

'c#', 'sql',

'c', 'c++',

'java', 'javascript',

'php', 'python',

'html/css', 'asp.net'

]

People who know javascript are :

Aniket Patra

Arya Dash

Kalyani Nayak

**10. Create a class Person with a constructor that takes name and age as parameters. Add a method getInfo that returns a string with the person's name and age. Implement inheritance by creating a class Student that extends the Person class. Add new properties like roll and grade to the Student class. Implement suitable methods to display student details.**

**Ans-**

//21BCSE30 | ANIKET PATRA

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

getInfo() {

return `Name: ${this.name}, Age: ${this.age}`;

}

}

class Student extends Person {

constructor(name, age, roll, grade) {

super(name, age);

this.roll = roll;

this.grade = grade;

}

displayDetails() {

console.log(`Name: ${this.name}`);

console.log(`Age: ${this.age}`);

console.log(`Roll: ${this.roll}`);

console.log(`Grade: ${this.grade}`);

}

}

const student1 = new Student("John", 20, 101, "A");

console.log(student1.getInfo());

student1.displayDetails();

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-3\ass3\q10.js"

Name: John, Age: 20

Name: John

Age: 20

Roll: 101

Grade: A