JavaScript Question Paper

Date:13/12/2024 **Time: 10 to 11.30 am**

Q.1) Write a function fetchData() that returns a promise. The promise should resolve after a 2-second delay, with the message "Data loaded". Once the promise resolves, the message should be logged to the console.

Function Name: fetchData

Return Type: The function should return a Promise.

Promise Behavior:

It should resolve with the string message "Data loaded" after 2 seconds (i.e., simulate some asynchronous data fetching).

When the Promise resolves: Log the message "Data loaded" to the console.

Q.2) Write a function parseJSON() that:

- Takes a JSON string as input.
- Returns the parsed JavaScript object if the JSON string is valid.
- If the JSON string is invalid (i.e., it cannot be parsed), catch the error and return the message "Invalid JSON".

⍰ **Function Name**: parseJSON

⍰ **Input**: A JSON-formatted string (e.g., '{ "name": "Ram", "age": 30 }').

⍰ **Output:**

- If the input string is a valid JSON, the function should return the parsed JavaScript object.
- If the input string is invalid (e.g., malformed JSON), the function should return the message "Invalid JSON".

⍰ **Error Handling:** Use try...catch to handle errors when attempting to parse invalid JSON.

Q.3) Write a function reverseString() that:

- Takes a string as input.
- Returns the string reversed.

Input:

- A string, e.g., "**hello JavaScript Developer**".

Output:

The string with each word reversed (e.g., "**olleh tpircSavaJ repoleveD**").

Q. 4) Write an arrow function groupBy() that:

- Takes an **array of objects** and a **property name** as arguments.
- Groups the objects by the specified property.
- Returns an **object** where the keys are the unique values of the specified property, and the values are **arrays** containing all objects that have that property value.
- **Input:**

```
 [
   { "name": "Amit", "age": 25 },{ "name": "Sanjay", "age": 25 },
   { "name": "Rajesh", "age": 30 },  { "name": "Priya", "age": 30 },
   { "name": "Neha", "age": 35 }
]
```

Output:

```
{
25: [ { "name": "Amit", "age": 25 },{ "name": "Sanjay", "age": 25 } ],
   30: [ { "name": "Rajesh", "age": 30 }, { "name": "Priya", "age": 30 }   ],
   35: [   { "name": "Neha", "age": 35 }   ]
}
```

Q. 5) Write a JavaScript function createTable() that:

Creates a table:

The table should have a **dynamic number of columns**, based on user input.

The table should have **3 rows** to start with, with each row containing input fields.

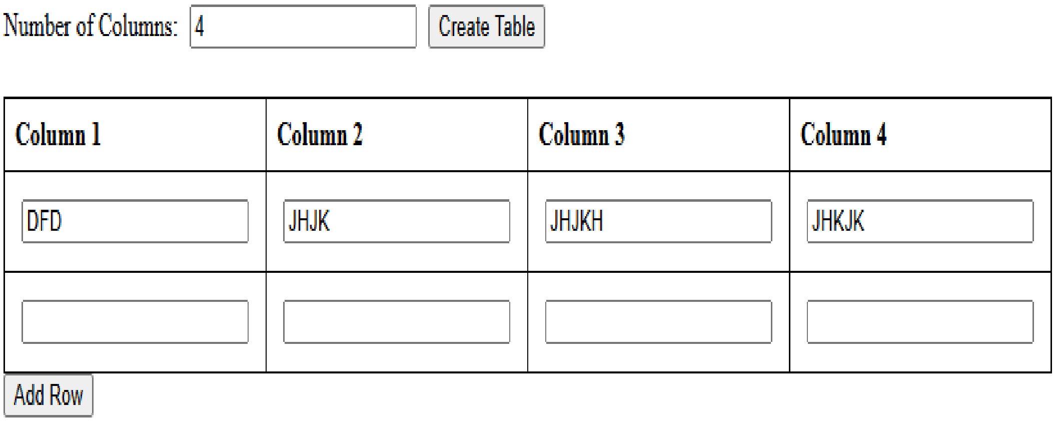
Input for Column Count:

The user should be able to specify the number of columns for the table via an input field.

Add Row Functionality:

When the user clicks an "Add Row" button, a new row should be added to the table.

Each new row should contain input fields, where the number of input fields corresponds to the number of columns specified by the user.



Number of Columns: 4

Column 1	Column 2	Column 3	Column 4
DFD	JHJK	JHJKH	JHKJK

Q.6) Write an arrow function mergeArrays that takes two arrays as input and returns a new array that is the concatenation of both arrays.

- **Example:**
  - **Input:** [1, 2], [3, 4]
  - **Output:** [1, 2, 3, 4]

--- All The Best ---