Homework 2-1 (in class)
Simple Shopping List Rewriting with Different Frameworks

Project Description

A simple “Shopping List” HTML, CSS, and JavaScript front-end web page that can be used as a basis for rewriting with React and Vue.js. Please try to use these two frameworks to rewrite the “Shopping List”.

Homework Submissions Requirement: (Choose one)

I.

1. Create a folder named “**student id_HW2**”. (ex: R12725088_HW2)
2. Each process should have its own folder.
3. A PDF document named README.pdf to explain what you have done in each process. (Attaching screenshots to help explain is recommended.)
4. Compressed the fold “student id_HW2” into a single zip file and uploaded it to COOL.

The folder should look like this:

```
R12725088_HW2
├─ README.PDF
├─ HW2_1
        ├─ Part 1
                ├─ index.html
                ├─ A.css
                ├─ B.js
        ├─ Part 2
                ├─ index.html
                ├─ A.css
                ├─ B.js
        ├─ Part 3
                ├─ index.html
                ├─ A.css
                ├─ B.js
├─ HW2_2
        ├─ Part 1
                ├─ index.html
                ├─ A.css
                ├─ B.js
        ├─ Part 2
                ├─ index.html
                ├─ A.css
                ├─ B.js
        ├─ Part 3
                ├─ index.html
├─ A.css
├─ B.js
```

II.

1. Create a folder named “**student id_HW2**”. (ex: R12725088_HW2)
2. A PDF document named README.pdf, including codepen’s link of each process, to explain what you have done in each process. (Attaching screenshots to help explain is recommended.)
3. Compressed the fold “student id_HW2” into a single zip file and uploaded it to COOL.

Given Frontend (HTML, CSS, and Javascript):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Shopping List</title>
  <style>
      body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        background-color: #f4f4f4;
      }

      .container {
        max-width: 600px;
        margin: 50px auto;
        padding: 20px;
        background-color: #fff;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      }

      h1 {
        text-align: center;
        margin-bottom: 20px;
      }

      input[type="text"] {
        width: calc(100% - 80px);
        padding: 10px;
        margin-bottom: 10px;
      }

      button {
        padding: 10px 20px;
        background-color: #007bff;
        color: #fff;
        border: none;
        border-radius: 5px;
        cursor: pointer;
      }

      button:hover {
        background-color: #0056b3;
```

```
            }

            ul {
                list-style-type: none;
                padding: 0;
            }

            li {
                padding: 10px;
                background-color: #f9f9f9;
                border-bottom: 1px solid #ddd;
                display: flex;
                justify-content: space-between;
            }

            li:last-child {
                border-bottom: none;
            }
    </style>
</head>
<body>
    <div class="container">
        <h1>Shopping List</h1>
        <input type="text" id="itemInput" placeholder="Add new item">
        <button onclick="addItem()">Add Item</button>
        <ul id="itemList"></ul>
    </div>
    <script>
            function addItem() {
                var input = document.getElementById("itemInput");
                var itemText = input.value.trim();
                if (itemText !== '') {
                    var itemList = document.getElementById("itemList");
                    var li = document.createElement("li");
                    li.textContent = itemText;
                    var deleteButton = document.createElement("button");
                    deleteButton.textContent = "Delete";
                    deleteButton.addEventListener("click", function() {
                        itemList.removeChild(li);
                    });
                    li.appendChild(deleteButton);
                    itemList.appendChild(li);
                    input.value = '';
                }
            }
    </script>
</body>
```

```
</html>
```

You can take this basic HTML and JavaScript code and rewrite it using two frameworks (React and Vue.js) according to their respective syntax and conventions. This will give you a starting point to learn and practice with each technology.

MUST (30%)

Part 1: Dismantle given code to three parts, html, css, and javascript (10%)

Please use a text editor or codepen website to do this.

Part 2: React.js (10%)

Please use React.js to rewrite the given frontend.

Part 3: Vue.js (10%)

Please use Vue.js to rewrite the given frontend.

Homework 2-2 (at home)
Weather App Rewriting with Different Frameworks

Project Description

A simple “Weather APP” HTML, CSS, and JavaScript front-end web page that can be used as a basis for rewriting with React and Vue.js. Please try to use these two frameworks to rewrite the “Weather APP”.

Homework Submissions Requirement:

Same as before.

Given Frontend (HTML, CSS, and Javascript):

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Weather App</title>
   <style>
        body {
           font-family: Arial, sans-serif;
           margin: 0;
           padding: 0;
           display: flex;
           justify-content: center;
           align-items: center;
           height: 100vh;
           background-color: #f0f0f0;
        }

        .container {
           text-align: center;
        }

        h1 {
           margin-bottom: 20px;
        }

        input[type="text"] {
           padding: 8px;
           margin-right: 5px;
        }
        button {
            padding: 8px 20px;
            background-color: #007bff;
            color: white;
            border: none;
            cursor: pointer;
        }

        #weather-info {
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Weather App</h1>
        <input type="text" id="city-input" placeholder="Enter city name">
        <button id="search-button">Search</button>
        <div id="weather-info"></div>
    </div>

    <script>
        document.getElementById("search-button").addEventListener("click",
        function() {
            var city = document.getElementById("city-input").value;
            if (city.trim() !== "") {
                fetchWeather(city);
            } else {
                alert("Please enter a city name.");
            }
        });

        function fetchWeather(city) {
            var apiUrl = `https://wttr.in/${city}?format=%t+%w+%h`;

            fetch(apiUrl)
                .then(response => {
                    if (!response.ok) {
                        throw new Error("Failed to fetch weather data.");
                    }
                    return response.text();
                })
                .then(data => {
                    displayWeather(data);
                })
                .catch(error => {
```

```
                console.error("Error fetching weather data:", error);
                alert("Failed to fetch weather data. Please try again later.");
            });
        }

        function displayWeather(data) {
            var weatherInfo = document.getElementById("weather-info");
            weatherInfo.innerText = data;
        }
    </script>
</body>
</html>
```

You can take this basic HTML and JavaScript code and rewrite it using two frameworks (React and Vue.js) according to their respective syntax and conventions. This will give you a starting point to learn and practice with each technology.

MUST (70%)

Part 1: Dismantle given code to three parts, html, css, and javascript (20%)

Please use a text editor or codepen website to do this.

Part 2: React.js (25%)

Please use React.js to rewrite the given frontend.

Part 3: Vue.js (25%)

Please use Vue.js to rewrite the given frontend.