

From the below url use any 3 api to implement the tasks with listed below constraints.(  
<https://github.com/public-apis/public-apis> )

Constaints:

- \*UI should be responsive &look and feel.
- \* use promise for retrieve data
- \* display the necessary things on the webpage
- \*your project /task should obtain the maximum code quality as below.
- \*proper variable function names
- \*use of catch for the fetch
- \* should not use deprecated tags
- \*should avoid inline styles
- \*should use only class selectors
- \* proper open &close of html tags
- \* comments /description of the function logic
- \*your project/task should follow code reusability
- \*use only html ,css,bootstrap,javascrpts

1)cat facts

Api: "https://alexwohlbruck.github.io/cat-facts/"

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cat Facts</title>
  <link rel="stylesheet" href="style2.css">
</head>
<body>
  <div class="container">
    <h1>Cat Facts</h1>
    <div class="cat-facts"></div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

### CSS CODE:

```
.container {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
}

.cat-facts {
  margin-top: 20px;
}

.cat-fact {
  margin-bottom: 10px;
}
```

### JAVASCRIPT CODE:

```
document.addEventListener("DOMContentLoaded", () => {
  const url = "https://alexwohlbruck.github.io/cat-facts/";

  // Fetch cat facts
  const fetchCatFacts = async () => {
    try {
      const response = await fetch(url);
      if (!response.ok) {
        throw new Error("Network response was not ok");
      }
      const data = await response.json();
      displayCatFacts(data);
    } catch (error) {
      console.error("Error fetching data:", error);
      displayErrorMessage("Failed to fetch cat facts. Please try again later.");
    }
  };

  // Display cat facts on the webpage
  const displayCatFacts = (catFacts) => {
    const catFactsContainer = document.querySelector(".cat-facts");
    if (!Array.isArray(catFacts)) {
      displayErrorMessage("Invalid data format. Please try again later.");
      return;
    }
    catFacts.forEach((fact) => {
      const factElement = document.createElement("div");
      factElement.classList.add("cat-fact");
```

```

        factElement.textContent = fact.text;
        catFactsContainer.appendChild(factElement);
    });
};

// Display error message
const displayErrorMessage = (message) => {
    const catFactsContainer = document.querySelector(".cat-facts");
    const errorMessage = document.createElement("div");
    errorMessage.classList.add("error-message");
    errorMessage.textContent = message;
    catFactsContainer.appendChild(errorMessage);
};

// Call fetchCatFacts function when the DOM content is loaded
fetchCatFacts();
});

```

## 2)DOG:

### HTML CODE:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dog Facts</title>
    <link rel="stylesheet" href="style2.css">
</head>
<body>
    <div class="container">
        <h1>Dog Facts</h1>
        <div class="dog-facts"></div>
    </div>

    <script src="script.js"></script>
</body>
</html>

```

### CSS CODE:

```

.container {

```

```

    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
  }

  .dog-facts {
    margin-top: 20px;
  }

  .dog-fact {
    margin-bottom: 10px;
  }

  .error-message {
    color: red;
  }

```

#### JAVASCRIPT CODE:

```

document.addEventListener("DOMContentLoaded", () => {
  const url = "https://dog.ceo/api/breeds/image/random/5";

  // Fetch dog facts
  async function fetchDogFacts() {
    try {
      const response = await fetch(url);
      if (!response.ok) {
        throw new Error("Network response was not ok");
      }
      return await response.json();
    } catch (error) {
      throw error;
    }
  }

  // Display dog facts on the webpage
  const displayDogFacts = (dogFacts) => {
    const dogFactsContainer = document.querySelector(".dog-facts");
    dogFacts.message.forEach((fact) => {
      const factElement = document.createElement("div");
      factElement.classList.add("dog-fact");
      factElement.textContent = fact;
      dogFactsContainer.appendChild(factElement);
    });
  };

  // Display error message

```

```

const displayErrorMessage = (message) => {
  const dogFactsContainer = document.querySelector(".dog-facts");
  const errorMessage = document.createElement("div");
  errorMessage.classList.add("error-message");
  errorMessage.textContent = message;
  dogFactsContainer.appendChild(errorMessage);
};

// Call fetchDogFacts function when the DOM content is loaded
fetchDogFacts()
  .then(data => {
    displayDogFacts(data);
  })
  .catch(error => {
    console.error("Error fetching data:", error);
    displayErrorMessage("Failed to fetch dog facts. Please try again
later.");
  });
});

```

### 3) SUSTAINABLE SEAFOOD:

#### HTML CODE:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NOAA Sustainable Seafood</title>
</head>
<body>

  <div class="container mt-5">
    <h1 class="text-center">NOAA Sustainable Seafood</h1>
    <div id="seafoodData" class="mt-4"></div>
  </div>

  <script src="script.js"></script>
</body>
</html>

```

#### JAVASCRIPT CODE:

```

document.addEventListener("DOMContentLoaded", async () => {
  const url = "https://www.fisheries.noaa.gov/topic/sustainable-seafood";

  async function fetchData(url) {

```

```

try {
  const response = await fetch(url);

  if (!response.ok) {
    throw new Error("Fetching data failed");
  }

  const data = await response.json();
  displaySeafoodData(data);
}
catch (error) {
  console.error("Error fetching data:", error);
}
}

function displaySeafoodData(data) {
  const seafoodDataContainer = document.getElementById("seafoodData");
  seafoodDataContainer.innerHTML = ""; // Clear previous data

  data.forEach(seafood => {
    const card = document.createElement("div");
    card.classList.add("card", "mb-3");

    const cardBody = document.createElement("div");
    cardBody.classList.add("card-body");

    const title = document.createElement("h5");
    title.classList.add("card-title");
    title.textContent = seafood.title;

    const description = document.createElement("p");
    description.classList.add("card-text");
    description.textContent = seafood.description;

    cardBody.appendChild(title);
    cardBody.appendChild(description);
    card.appendChild(cardBody);

    seafoodDataContainer.appendChild(card);
  });
}

// Call fetchData function when the DOM content is loaded
fetchData(url);
});

```

