Practical – 10

Aim:

Source Code:

1) **Dogs and Cats API in Async/Await mode.**

HTML Code:-

```
<p>Getting images of doges and cats from fetching Apis in Async/Await Mode.</p>
<div class=”results”>
    <div id=”cat_result” class=”result”>
        <p>Random Cat Placeholder</p>
    </div>
    <div id=”dog_result” class=”result”>
        <p>Random Dog Placeholder</p>
    </div>
</div>

<div class=”buttons”>
    <button id=”cat_btn”>Get Cat</button>
    <button id=”dog_btn”>Get Dog</button>
</div>
```

CSS Code:-

```
@import url(‘https://fonts.googleapis.com/css?family=Open+Sans&display=swap’);
Body {
    Background-image: linear-gradient( 180deg,  rgba(247,247,247,1) 23.8%, 
rgba(252,221,221,1) 92% );
    Font-family: ‘Open Sans’, sans-serif;
    Display: flex;
    Align-items: center;
    Justify-content: center;
    Flex-direction: column;
    Height: 100vh;
    Margin: 0;
}

.results {
    Display: flex;
    Align-items: center;
    Justify-content: center;
}

.result {
    Background-color: #fff;
```

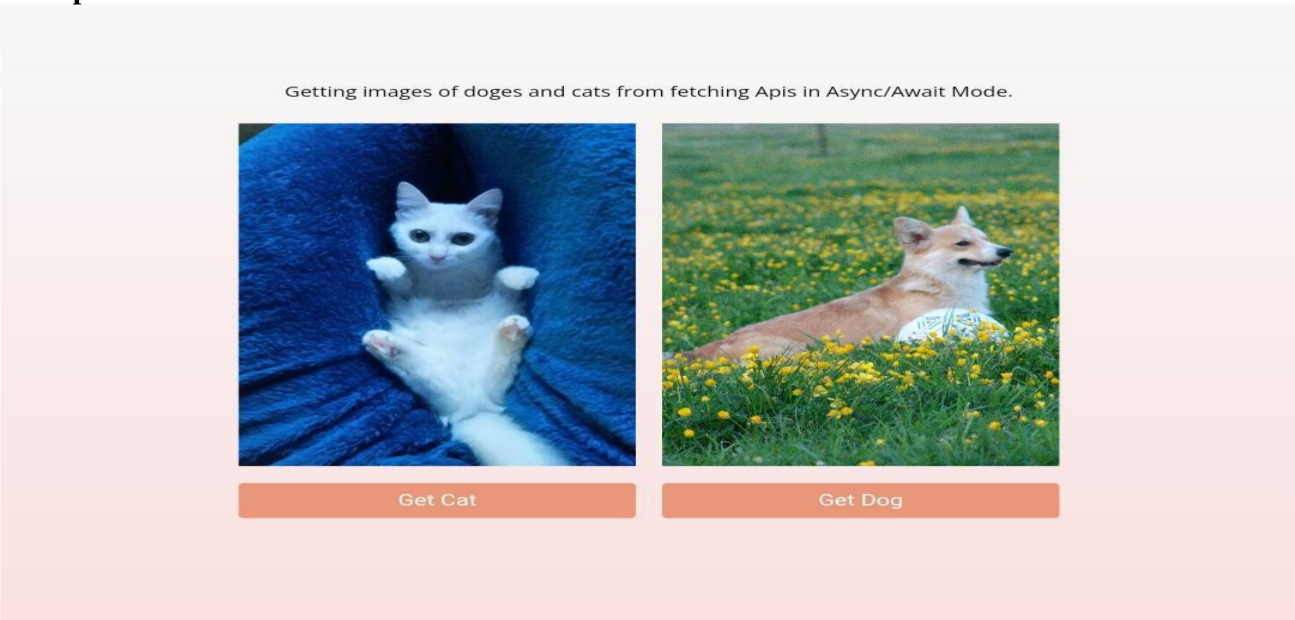
```
        Display: flex;
        Align-items: center;
        Justify-content: center;
        Margin: 10px;
        Height: 400px;
        Width: 300px;
}

.result img {
        Object-fit: cover;
        Height: 100%;
        Width: 100%;
}

.buttons {
        Display: flex;
        Align-items: center;
        Justify-content: center;
}

Button {
        Background-color: darksalmon;
        Border: 0;
        Border-radius: 4px;
        Color: #fff;
        Font-size: 18px;
        Padding: 10px;
        Margin: 10px;
        Width: 300px;
}

Button:active {
        Transform: scale(0.98);
}

Button:focus {
        Outline: none
}
```

JavaScript Code:-

```
Const cat_btn = document.getElementById(‘cat_btn’);
Const dog_btn = document.getElementById(‘dog_btn’);
Const cat_result = document.getElementById(‘cat_result’);
Const dog_result = document.getElementById(‘dog_result’);
```

```
Cat_btn.addEventListener(‘click’, getRandomCat);
Dog_btn.addEventListener(‘click’, getRandomDog);
Function getRandomCat() {
        Fetch(‘https://aws.random.cat/meow’)
                .then(res => res.json())
                .then(data => {
                        Cat_result.innerHTML = `<img src=${data.file} alt=”cat” />`
                });
}
Function getRandomDog() {
        Fetch(‘https://random.dog/woof.json’)
                .then(res => res.json())
                .then(data => {
                        If(data.url.includes(‘.mp4’)) {
                                getRandomDog();
                        }
                        Else {
                                Dog_result.innerHTML = `<img src=${data.url} alt=”dog” />`;
                        }
                });
}
```

Output:-



Getting images of doges and cats from fetching Apis in Async/Await Mode.

2)

HTML Code:-

```
<div id=”cocktail”></div>    <div id=”overlay”></div>    <script src=”script.js”></script>
```

CSS Code:-

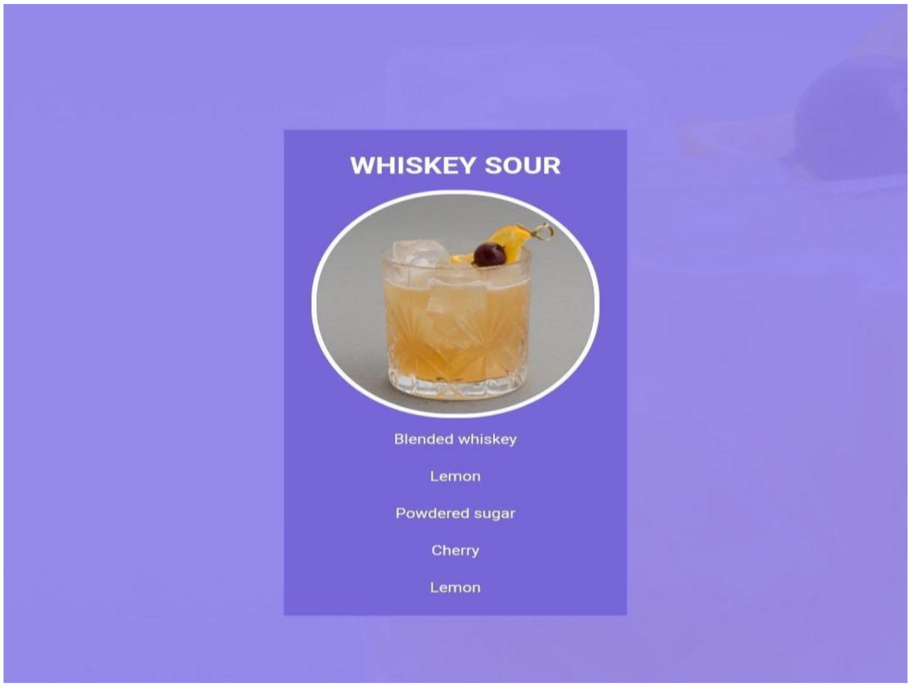
```
Html {   height: 100%; }
 body {   display: flex;   justify-content: center;   align-items: center;   height: 100%;
background-size: cover;   font-family: sans-serif; }
 #overlay {   background: rgba(147, 135, 242, 0.9);   position: fixed;   top: 0;   left: 0;   width:
100%;   height: 100%;   z-index: -1; }
 #cocktail {   max-width: 350px;   text-align: center;   padding: 30px 30px 12px 30px;   color:
#fff;   background-color: #7766d7;   border: 4px solid #9387f2;   border-radius: 5px; }
#cocktail h1 {   margin: 0 0 15px 0;   text-transform: uppercase; }
#cocktail img {   max-width: 300px;   border: 6px solid #fff;   border-radius: 150px; }
 #cocktail ul {   list-style: none;   margin: 0;   padding: 0; }
 #cocktail li {   padding: 15px 0;   font-size: 18px; }
#cocktail li:not(:last-of-type) {   border-bottom: 1px solid #fff; }
```

JavaScript Code:-

```
fetch("https://www.thecocktaildb.com/api/json/v1/1/random.php")
  .then((response) => {
    if (response.ok) {
      return response.json();
    } else {
      throw new Error("NETWORK RESPONSE ERROR");
    }
  })
  .then(data => {
    console.log(data);
    displayCocktail(data)
  })
  .catch((error) => console.error("FETCH ERROR:", error));

function displayCocktail(data) {
  const cocktail = data.drinks[0];
  const cocktailDiv = document.getElementById("cocktail");
  // cocktail name
  const cocktailName = cocktail.strDrink;
  const heading = document.createElement("h1");
  heading.innerHTML = cocktailName;
  cocktailDiv.appendChild(heading);
  // cocktail image
  const cocktailImg = document.createElement("img");
  cocktailImg.src = cocktail.strDrinkThumb;
  cocktailDiv.appendChild(cocktailImg);
  document.body.style.backgroundImage = "url('" + cocktail.strDrinkThumb + "')";
  // cocktail ingredients
  const cocktailIngredients = document.createElement("ul");
  cocktailDiv.appendChild(cocktailIngredients);
  const getIngredients = Object.keys(cocktail)
```

```
    .filter(function (ingredient) {
      return ingredient.indexOf("strIngredient") == 0;
    })
    .reduce(function (ingredients, ingredient) {
      if (cocktail[ingredient] != null) {
        ingredients[ingredient] = cocktail[ingredient];
      }
      return ingredients;
    }, {});
  for (let key in getIngredients) {
    let value = getIngredients[key];
    listItem = document.createElement("li");
    listItem.innerHTML = value;
    cocktailIngredients.appendChild(listItem);
  }
}
```

Output:-



WHISKEY SOUR

Blended whiskey

Lemon

Powdered sugar

Cherry

Lemon

3)

HTML Code:-

```
<!DOCTYPE html>
```

```
<html lang=”en”>

   <head>

      <script src=”script.js”></script>
      <link rel=”stylesheet” href=”style.css” />
      <meta charset=”UTF-8” />
      <meta name=”viewport”
           Content=”width=device-width, initial-scale=1.0” />
      <title>Document</title>
   </head>

   <body>
      <!—Here a loader is created which
          Loads till response comes ⍰
      <div class=”d-flex justify-content-center”>

         <div class=”spinner-border”
             Role=”status” id=”loading”>
            <span class=”sr-only”>Loading…</span>
         </div>

      </div>
      <h1>Registered Employees</h1>
      <!—table for showing data ⍰
      <table id=”employees”></table>
   </body>

</html>
```

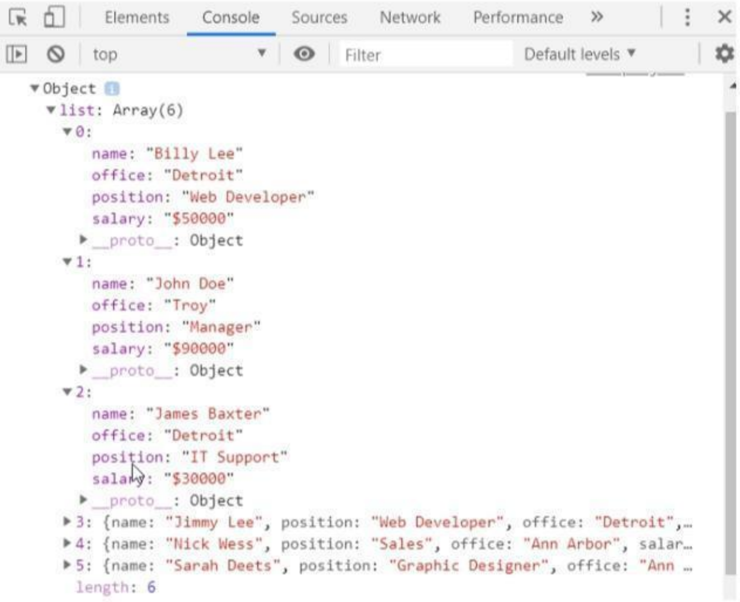
JavaScript Code:-

```
// api url
Const api_url =   https://employeedetails.free.beeceptor.com/my/api/path;

// Defining async function
Async function getapi(url) {
   // Storing response
   Const response = await fetch(url);
   // Storing data in form of JSON
   Var data = await response.json();
   Console.log(data);
   If (response) {
      Hideloader();
```

```
    }
    Show(data);
}

// Calling that async function
Getapi(api_url);
// Function to hide the loader
Function hideloader() {
    Document.getElementById('loading').style.display = 'none';
}

// Function to define innerHTML for HTML table
Function show(data) {
    Let tab =
        `<tr>
          <th>Name</th>
          <th>Office</th>
          <th>Position</th>
          <th>Salary</th>
         </tr>`;

    // Loop to access all rows
    For (let r of data.list) {
        Tab += `<tr>
    <td>${r.name} </td>
    <td>${r.office}</td>
    <td>${r.position}</td>
    <td>${r.salary}</td>
</tr>`;
    }

    // Setting innerHTML as tab variable
    Document.getElementById("employees").innerHTML = tab;
}
```

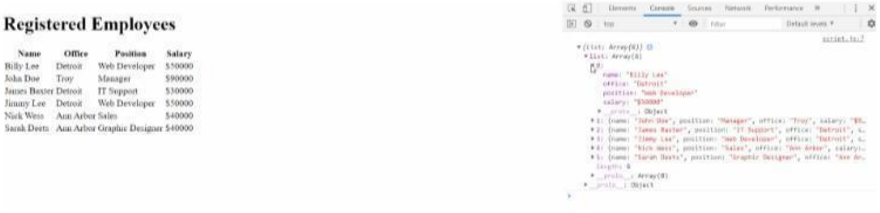
Output:-

- In the console, data in JSON will look like this.



```
▼Object
 ▼list: Array(6)
  ▼0:
     name: "Billy Lee"
     office: "Detroit"
     position: "Web Developer"
     salary: "$50000"
    ▶__proto__: Object
  ▼1:
     name: "John Doe"
     office: "Troy"
     position: "Manager"
     salary: "$90000"
    ▶__proto__: Object
  ▼2:
     name: "James Baxter"
     office: "Detroit"
     position: "IT Support"
     salary: "$30000"
    ▶__proto__: Object
  ▶3: {name: "Jimmy Lee", position: "Web Developer", office: "Detroit",…
  ▶4: {name: "Nick Wess", position: "Sales", office: "Ann Arbor", salar…
  ▶5: {name: "Sarah Deets", position: "Graphic Designer", office: "Ann …
   length: 6
```

- HTML Output.



Registered Employees

Name	Office	Position	Salary
Billy Lee	Detroit	Web Developer	$50000
John Doe	Troy	Manager	$90000
James Baxter	Detroit	IT Support	$30000
Jimmy Lee	Detroit	Web Developer	$50000
Nick Wess	Ann Arbor	Sales	$40000
Sarah Deets	Ann Arbor	Graphic Designer	$40000

```
▼{list: Array(6)}
 ▼list: Array(6)
  ▼0:
     name: "Billy Lee"
     office: "Detroit"
     position: "Web Developer"
     salary: "$50000"
    ▶__proto__: Object
  ▶1: {name: "John Doe", position: "Manager", office: "Troy", salary: "$9…
  ▶2: {name: "James Baxter", position: "IT Support", office: "Detroit", s…
  ▶3: {name: "Jimmy Lee", position: "Web Developer", office: "Detroit", s…
  ▶4: {name: "Nick Wess", position: "Sales", office: "Ann Arbor", salary:…
  ▶5: {name: "Sarah Deets", position: "Graphic Designer", office: "Ann Ar…
   length: 6
  ▶__proto__: Array(0)
 ▶__proto__: Object
>
```

Conclusion:

- I learned how to fetch different APIs in Asyc/Await mode in JavaScript.

Course Outcome:

- I learned how to make a responsive website which not only takes values or inputs from user but also returns the answer or output by calculating in backend which user doesn't need to know.