```
<script type="text/javascript">
    const userAPI = "https://jsonplaceholder.typicode.com/users/1";
    const postsAPI = "https://jsonplaceholder.typicode.com/posts?userId=1";

    const fetchUserDataAndPosts = async () => {
    try {
      const response1 = await fetch(userAPI);
      const response2 = await fetch(postsAPI);

      if (response1.ok && response2.ok) {
      const userData = await response1.json();
      const postsData = await response2.json();
      return { user: userData, posts: postsData };
      } else {
      throw new Error("Server Error");
      }
    } catch (err) {
      console.error("Error fetching data", err);
      throw err;
    }
    }

    (async () => {
    try {
      const { user, posts } = await fetchUserDataAndPosts();
      console.log("User:", user);
      console.log("Posts:", posts);
    } catch (err) {
      console.error("Error fetching data", err);
    }
    })();
</script>
```

```
<script type="text/javascript">

    const url07 = "https://jsonplaceholder.typicode.com/posts";
    const url03 = "https://jsonplaceholder.typicode.com/photos";

    const fetchData = () => {
      const apiUrls = [url03,url07];

      const promises = apiUrls.map(url=>{
        return fetch(url)
        .then(response=>{
          if(response.ok) {
            return response.json();
          }
          throw new Error("Server Error");
        })
        .catch(err =>{
          console.error("Data not received",err);
        })
      })

      return Promise.all(promises)
      .then(data=>{
        return data;
      })
    }


    fetchData()
    .then(data=>console.log(data))
    .catch(err=>console.error("Data not Received",err) )
    // SAME USING ASYNC AWAIT

    const fetchDataUsingAsync = async () => {
      const apiUrls = [url03,url07];

      try{
        const responses = await Promise.all(apiUrls.map(url=>fetch(url)));
        const dataArray = await Promise.all(responses.map(response=>{
          if(response.ok){
            return response.json()
          }
          throw new Error("Server not connected")
        }))

        return dataArray;
      }catch(err){
        console.log(err.message);
        throw err;
      }

    }


    (async() => {
      try{
        let data = await fetchDataUsingAsync();
        console.log(data);
      }catch(err){
        console.log(err.message);
      }
    })();

  </script>
```