

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Run. Below that are sections for 'OPEN EDITORS' (JS Fetch.js, JS Async.js) and 'TASK 4' (JS Async.js, Fetch.js). The main area has tabs for 'JS Fetch.js' and 'JS Async.js'. The code editor displays the following JavaScript code:

```
1 //this function fetches data from a url
2 function fetchRetry(url, retries = 5){ //if req fails ,it retries again 3times
3     return fetch(url) //call the fetch Api with give url
4
5     .then(response =>{ //handle response
6         if(!response.ok){ //if url status is not sucessful
7             throw new Error("Request failed"); //throw Error
8         }
9         return response.json(); //convert response to json
10    })
11
12    .catch(error =>{ //catch error from above block
13        if(retries > 0){
14            console.log("retrying..."); //log message
15            return fetchRetry(url, retries -1) //call the same function again with retries reduced by 1
16        }
17        throw error;
18    })
19
20
21
22 // fetch('https://jsonplaceholder.typicode.com/todos/1') //correct url
23 fetchRetry('https://jsonplaceholder.typicode.com/to/invalid') //wrong url
24 .then(response => response.json()) //then runs when server responds successfully and response is response object not actual data
25 .then(json => console.log(json)) //response.json() converts body into javascript object and json contains actual api
26
27 .catch(error => console.log("Final Error: ", error.message));
```

To the right of the code editor, there's a 'Build with Agent' section with a button to 'Generate Agent Instructions'. Below it are 'SUGGESTED ACTIONS' buttons for 'Build Workspace' and 'Show Config'. At the bottom, there's a taskbar with various icons and a system tray showing the date and time.

The screenshot shows a code editor interface with multiple tabs and a sidebar. The main area displays a file named 'JS Asyncjs' containing the following JavaScript code:

```
function asyncTask(name, time, shouldFail=false) {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if(shouldFail) {
        |  reject(new Error(name + " failed"));
      }
      else{
        console.log(name + " completed"); //print when task is completed
        resolve(name);
      }
    },time);
  })
}

async function runSerial() { //serial task run one after another
  try {
    await asyncTask("task 1" , 2000);
  } catch(e) {
    console.log("task 1:", e.message);
  }

  try {
    //await asyncTask("task 2" , 2000); //error
    await asyncTask("task 2" , 2000);
  } catch (e) {
    |  console.log("task 2:", e.message);
  }
  try {
    await asyncTask("task 3" , 2000);
  } catch(e) {
    |  console.log("task 3:", e.message);
  }
  console.log("serial execution")
};

async function runParallel() {
  const tasks = [
    asyncTask("Task 4", 5000).catch(e => console.log("task 4: ", e.message)),
    asyncTask("Task 5", 7000,true).catch(e => console.log("task 5: ", e.message)),
    asyncTask("Task 6", 8000).catch(e => console.log("task 6: ", e.message))
  ];
}
```

The sidebar on the left lists 'OPEN EDITORS' with files 'JS Fetchjs', 'JS Asyncjs', and 'JS Fetchjs'. It also has sections for 'TASK 4' and 'OUTLINE/TIMELINE'. The bottom right corner shows a weather widget indicating 19°C Partly cloudy at 6:16 PM on 12/29/2025.

The screenshot shows a code editor window with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Task 4
- Editor Area:** The main area displays a file named "JS Asyncjs". The code is as follows:

```
17  async function runSerial() { //serial task run one after another
18      await asyncTask("task 1", 2000);
19  }
20  } catch(e) {
21      console.log("task 1:", e.message);
22  }
23  try {
24      //await asyncTask("task 2", 2000, true); //error
25      await asyncTask("task 2", 2000);
26  } catch (e) {
27      | console.log("task 2:", e.message);
28  }
29  try {
30      await asyncTask("task 3", 2000);
31  } catch(e) {
32      | console.log("task 3:", e.message);
33  }
34  console.log("serial execution")
35 };
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
```

Right Panel: Contains a "Build with Agent" section with a button to "Generate Agent Instructions" and a "SUGGESTED ACTIONS" section with "Build Workspace" and "Show Config" buttons.

Bottom Status Bar: In 38, Col 8, Spaces:4, UTF-8, CRLF, JavaScript, Go Live, Prettier.