Assignment 5a

Aim

Write a Javascript program.
a. Implement the concept of Promise(callback)
b. Fetch (Client Server Communication)

Theory

The concept of a Promise in JavaScript is a crucial mechanism for managing asynchronous operations in a structured and organized manner. When dealing with tasks that may take some time to complete, such as fetching data from a server or reading files, Promises offers a clear and predictable way to handle the flow of execution.

Promises operate based on several core principles

Asynchronous Operations: Promises are primarily designed to handle asynchronous tasks. These are operations that don't block the main program flow and may be completed at an unspecified time in the future. Common examples include making network requests, processing user input, or interacting with external resources.

States: Promises have three primary states:

- Pending: This is the initial state when a Promise is created. It signifies that the asynchronous operation has not yet been completed.
- Fulfilled (Resolved): In this state, the operation was successful, and the Promise holds a result or a value.
- Rejected: If the operation encounters an error or fails, the Promise enters the rejected state, and it contains a reason for the failure.
- Chaining: Promises support a chaining mechanism. Developers can use the .then() method to specify what should happen when a Promise resolves successfully. Chaining multiple .then() methods enables the creation of a sequence of actions that execute in a specific order.

Error Handling: Promises provide a structured approach to error handling through the .catch() method. If any error occurs within the Promise chain, it propagates to the nearest .catch() block, allowing for consistent error handling and logging.

Fetch API

The fetch() function in JavaScript is a method employed to initiate network requests and retrieve resources, such as data or files, from a server or another location on the internet. It is commonly used for making asynchronous HTTP requests. fetch() is considered a modern and more versatile alternative to the older XMLHttpRequest (XHR) for handling data fetching in web applications.

Key features and characteristics of the fetch() function include:

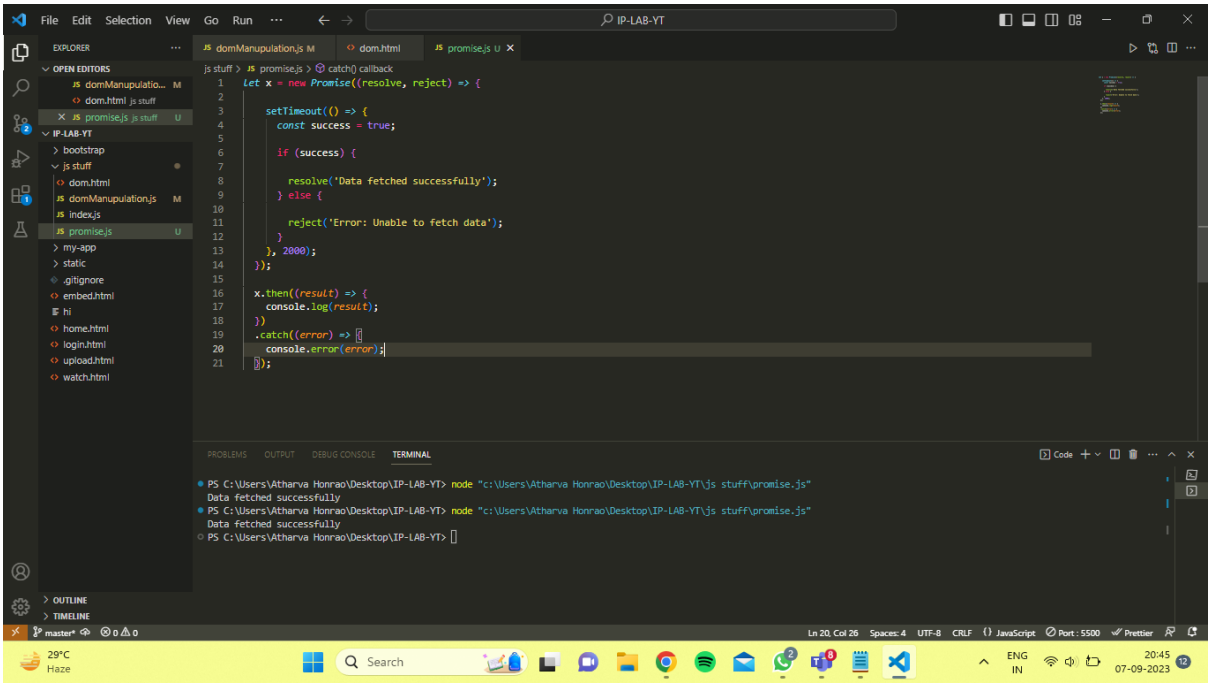
Asynchronous Requests: fetch() performs asynchronous operations, meaning it doesn't block the main execution thread of a web page. This allows web applications to continue functioning while waiting for a response from the server.

Promise-Based: fetch() returns a Promise, making it compatible with modern JavaScript's promise-based syntax. Promises provide a structured way to handle the result of asynchronous operations, whether they succeed or encounter errors.

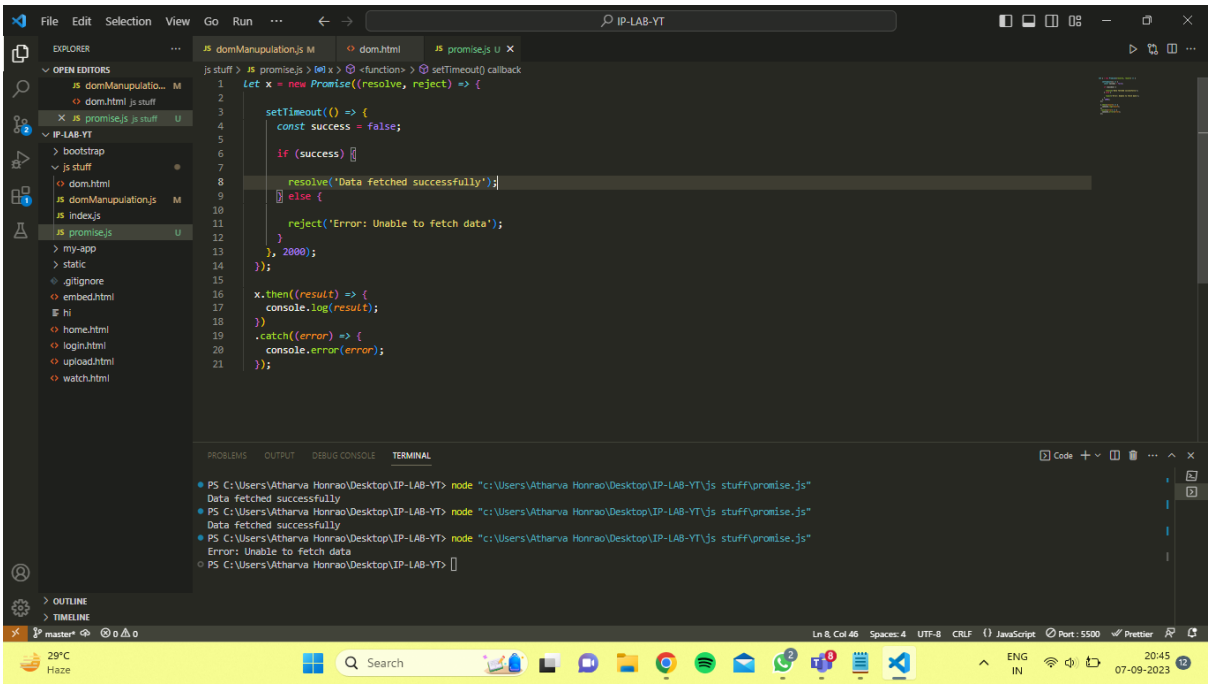
Atharva Honrao T13 - 43

Output Image

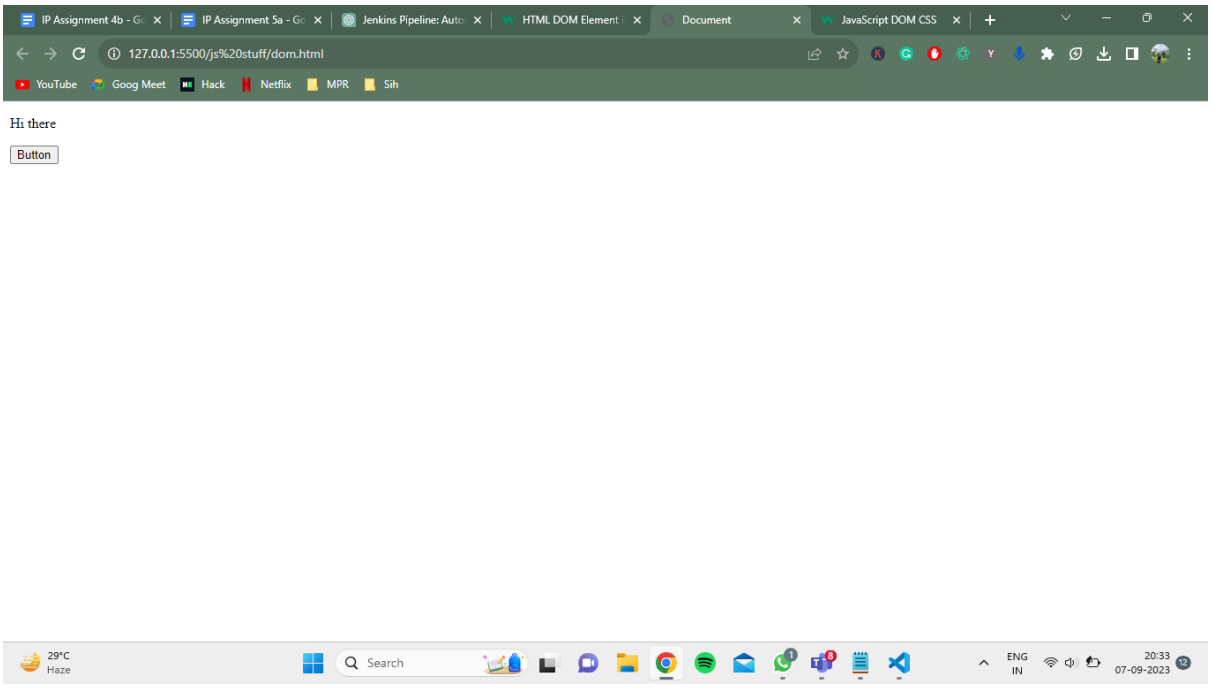
Resolve

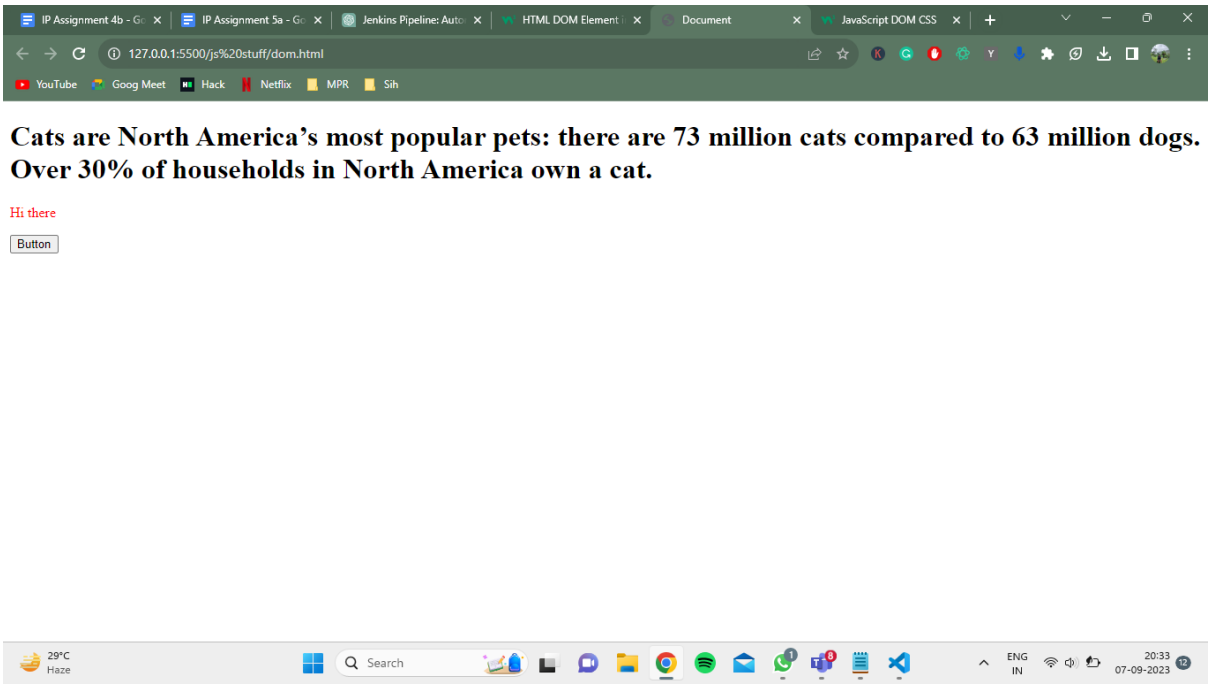


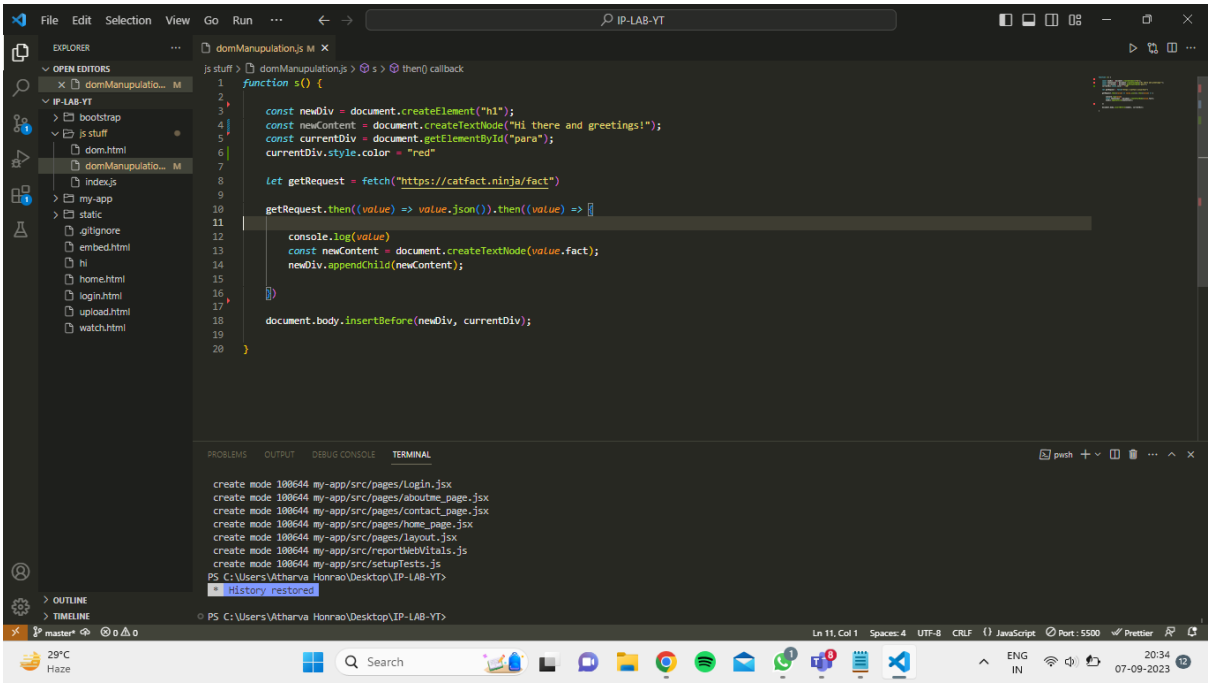
Reject



Fetch API


IP Assignment 4b - Go
IP Assignment 5a - Go
Jenkins Pipeline: Auto
HTML DOM Element
Document
JavaScript DOM CSS
127.0.0.1:5500/js%20stuff/dom.html
YouTube
Goog Meet
Hack
Netflix
MPR
Sih
Hi there
Button
29°C
Haze
Search
ENG
IN
20:33
07-09-2023


IP Assignment 4b - Go
IP Assignment 5a - Go
Jenkins Pipeline: Auto
HTML DOM Element
Document
JavaScript DOM CSS
127.0.0.1:5500/js%20stuff/dom.html
YouTube
Goog Meet
Hack
Netflix
MPR
Sih
Cats are North America's most popular pets: there are 73 million cats compared to 63 million dogs. Over 30% of households in North America own a cat.
Hi there
Button
29°C
Haze
Search
ENG
IN
20:33
07-09-2023



Conclusion

Promises in JavaScript are a fundamental concept for managing asynchronous operations. They provide a structured and reliable way to work with asynchronous tasks, enabling developers to handle both successful resolutions and errors in a more organized manner. The fetch() function in JavaScript is a modern and versatile method for making asynchronous network requests, commonly used for retrieving data from servers or other sources over HTTP.