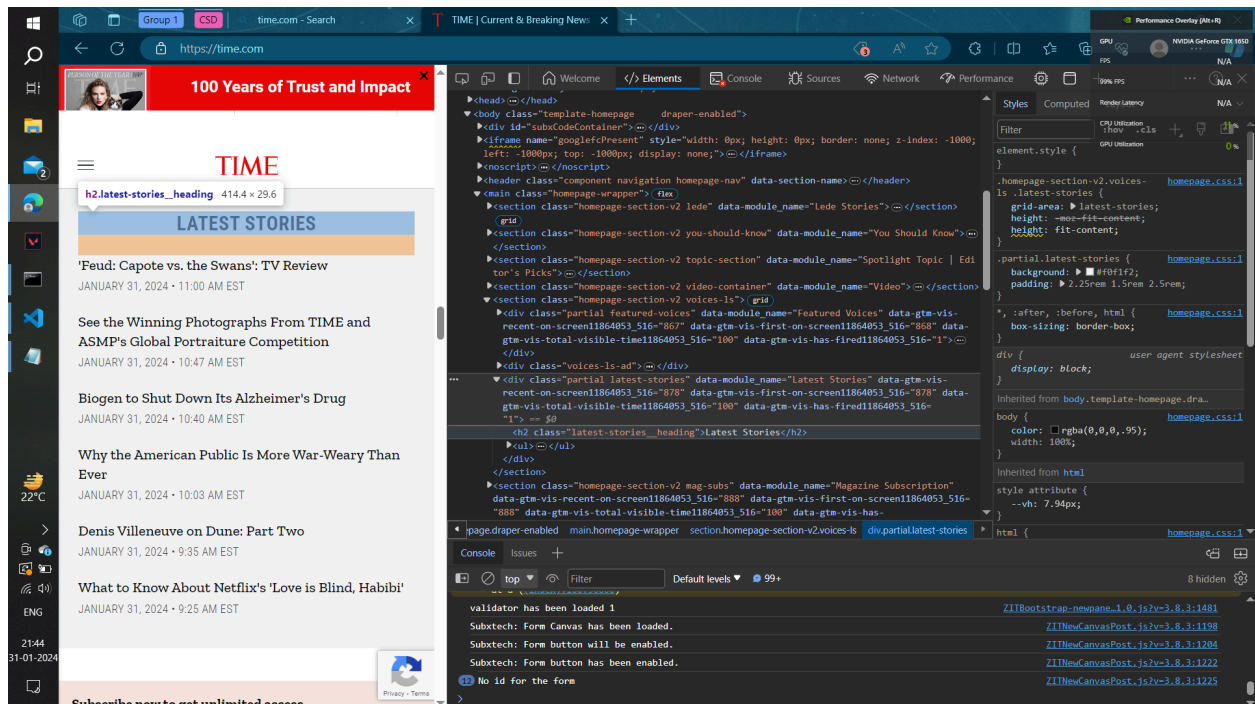


Project Report: Creating a Simple Server and API Endpoint for Time Stories



Objective: The goal of this project was to create a basic server and API endpoint without using external libraries or npm packages. The API endpoint `/getTimeStories` aims to fetch and present data from the Time website.

Server Creation: We chose Node.js as the runtime environment to create a server. The built-in `http` module was utilized for handling HTTP requests. The server was set up to listen on a specified port (in this case, 3000) and respond to requests. The server distinguishes between different routes and methods to provide appropriate responses.

Handling Routes and Methods: Routing and handling methods were managed through straightforward conditional statements. For example, when a GET request is made to the `/api` endpoint, the server issues a predetermined JSON object as a response. The implemented logic guarantees that the server adeptly manages various routes and methods, delivering a 404 status for unfamiliar routes. In the process of obtaining data from Time.com, the server executes a GET request to `https://time.com` utilizing the

built-in http module. It's important to recognize that this method might not effectively capture dynamically loaded content and might not be universally applicable to all scenarios.

Parsing HTML Content: Parsing HTML content was accomplished by utilizing regular expressions (regex). Specifically crafted regex patterns were employed to identify and match distinct HTML elements, facilitating the extraction of essential data such as titles and links. This strategy was adopted in accordance with the project limitations, which prohibited the use of third-party libraries for parsing.

Result image:

