**WAD ASSIGNMENT**

**Name:** Kaveri Waghchoure

**Roll no:** 69

**Package name:** Async.js

**Illustrate the need for Async.js** **Package:**

The async.js package is a powerful utility module for JavaScript that provides functions to handle asynchronous operations in a more organized and efficient manner. Here are some illustrations of the need for async.js in JavaScript applications:

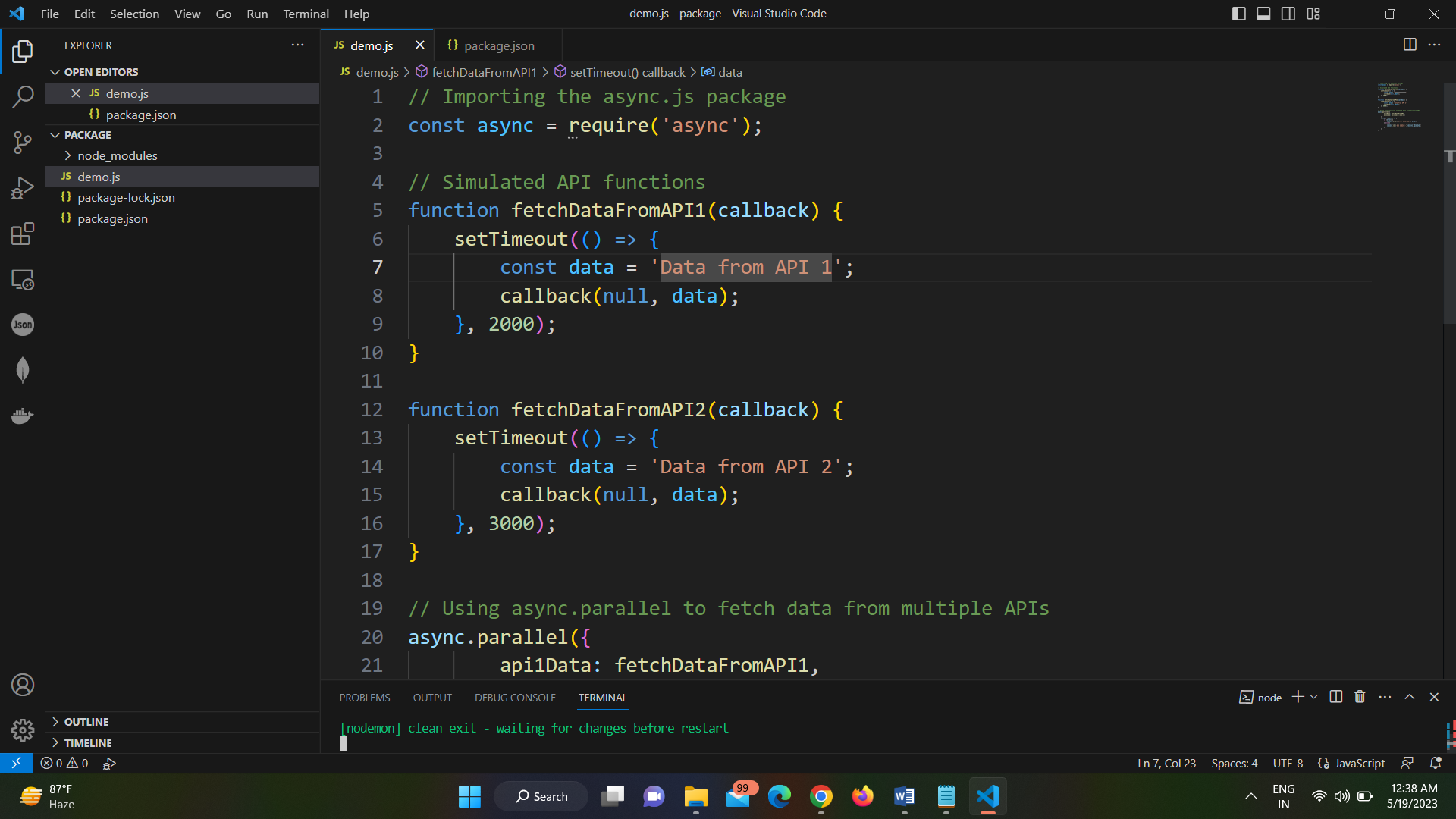
1. Managing Callback Hell: In JavaScript, when dealing with multiple asynchronous operations, the code can quickly become nested and hard to read, leading to a phenomenon known as "Callback Hell." async.js helps alleviate this issue by providing functions such as async.series, async.parallel, and async.waterfall that simplify the handling of asynchronous tasks, reducing the nesting and improving code readability.
2. Error Handling: Asynchronous operations can often encounter errors that need to be handled appropriately. Async.js provides mechanisms to handle errors in asynchronous code effectively. For instance, the async.waterfall function allows passing errors to the callback, ensuring that errors are propagated and can be handled centrally.
3. Parallel Execution: In scenarios where multiple asynchronous operations can be executed simultaneously, async.js offers the async.parallel function. This function executes multiple tasks in parallel, allowing for optimized performance when independent operations can run concurrently.
4. Asynchronous Iteration: The async.js library includes functions such as async.each and async.map that simplify iterating over arrays or collections asynchronously. These functions handle the iteration and invocation of asynchronous operations on each element, providing a concise and organized approach to asynchronous iteration.
5. Improved Code Readability: By providing a structured and organized approach to handling asynchronous tasks, async.js enhances code readability. It abstracts away the complexities of asynchronous programming, allowing developers to focus on the logic of their application rather than dealing with intricate callback structures.

**Implement the small application which will make the use of mentioned package:**

First install package: npm i async.js

In this example, we have two simulated API functions fetchDataFromAPI1 and fetchDataFromAPI2. Each function has a callback parameter that is called after a delay of 2 seconds and 3 seconds, respectively. These functions simulate asynchronous API calls.

**Demo.js:**



**Illustrate the need for code of ethics:**

The need for a code of ethics arises from several important considerations. Here are some key illustrations of why a code of ethics is necessary:

1. Guidance for Decision Making: A code of ethics provides individuals and organizations with a framework and principles to guide their decision-making processes. It establishes a set of standards and values that help individuals navigate ethical dilemmas and make choices aligned with ethical principles.
2. Establishing Trust and Credibility: A code of ethics helps build trust and credibility among stakeholders, including customers, clients, employees, and the general public. By adhering to a code of ethics, individuals and organizations demonstrate their commitment to ethical conduct, which enhances their reputation and fosters trust in their actions and operations.
3. Setting Expectations and Accountability: A code of ethics outlines the expected behaviour and conduct for individuals within an organization or profession. It establishes clear expectations, rules, and responsibilities, ensuring that individuals are aware of the ethical standards they are expected to uphold. This promotes accountability and provides a basis for evaluating and addressing ethical breaches.
4. Protecting Stakeholder Interests: A code of ethics helps protect the interests and well-being of various stakeholders. It ensures that individuals and organizations prioritize the rights, safety, and needs of stakeholders over personal or organizational gain. By following ethical principles, potential harm or exploitation of stakeholders can be minimized or prevented.
5. Safeguarding Integrity and Professionalism: A code of ethics upholds and preserves the integrity and professionalism of individuals and organizations. It promotes honest and transparent behaviour, discourages unethical practices, and encourages individuals to act in a manner consistent with the highest professional and moral standards.
6. Compliance with Laws and Regulations: A code of ethics often incorporates legal and regulatory requirements, ensuring that individuals and organizations operate within the boundaries of the law. It provides guidance on legal and ethical obligations, helping individuals understand and fulfil their legal responsibilities while maintaining ethical integrity.
7. Fostering a Positive Organizational Culture: A code of ethics contributes to the development of a positive organizational culture characterized by integrity, respect, and fairness. It sets the tone for ethical behaviour and serves as a unifying force, aligning individuals and teams towards a shared ethical purpose.
8. Adapting to Changing Contexts and Challenges: A code of ethics can be a dynamic document that evolves with changing social, cultural, and technological landscapes. It allows for the consideration and inclusion of emerging ethical issues and challenges, ensuring that individuals and organizations adapt and respond ethically to new circumstances