2021-22

Monday 9 January 2023 18:31

1 (a) You were required to design a Rich Web Application for a university. Define, with your own words, the term Rich Web Application and explain the differences between web applications and websites. List and describe 2 (two) features of the university rich web application you would develop and what types of technologies you would use to implement each of these features.

(10 marks)

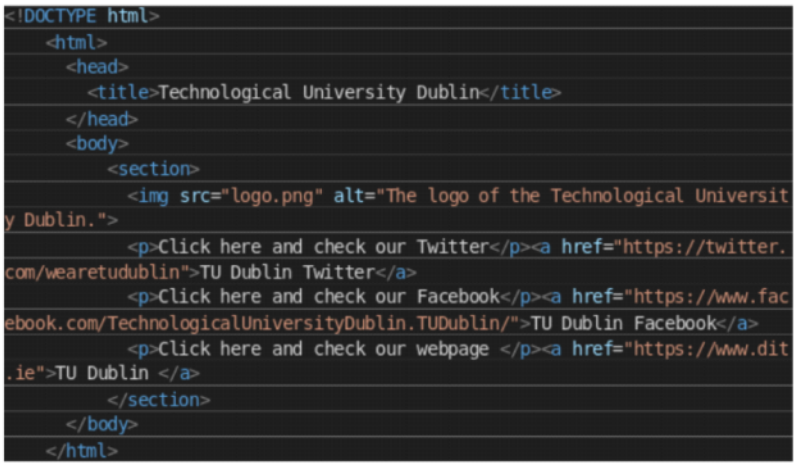
A Rich Web Application (RWA) is a type of web application that provides a more interactive and engaging user experience compared to a traditional website. RWAs often utilize modern web technologies such as AJAX, WebSockets, and HTML5 to create a seamless and responsive user interface.

One key difference between web applications and websites is that web applications are more interactive and dynamic, while websites are typically more static and content-based. Web applications often require a more complex technical architecture and may involve the use of databases and server-side logic, while websites are typically built using only client-side technologies such as HTML, CSS, and JavaScript.

For the university RWA, two features I would develop are:

1. A Course Scheduling System: This feature would allow students to view and enroll in available courses, as well as view their current schedule and any conflicts. To implement this feature, I would use technologies such as JavaScript, HTML, and CSS for the front-end user interface, and a server-side language such as PHP or Python to handle the back-end logic and communication with a database to store and retrieve course and enrollment data.
2. A Student Grades Portal: This feature would allow students to view their current grades and academic progress, as well as access past assignments and grades. To implement this feature, I would use technologies such as JavaScript, HTML, and CSS for the front-end user interface, and a server-side language such as PHP or Python to handle the back-end logic and communication with a database to store and retrieve student grade data.

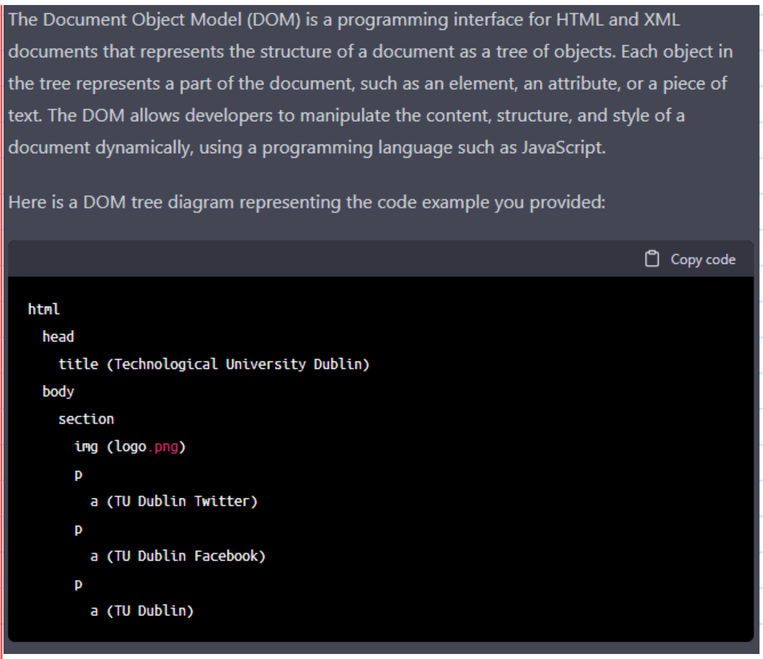
(b) Consider the following code example:



```html
<!DOCTYPE html>
    <html>
      <head>
        <title>Technological University Dublin</title>
      </head>
      <body>
          <section>
            <img src="logo.png" alt="The logo of the Technological Universit
y Dublin.">
            <p>Click here and check our Twitter</p><a href="https://twitter.
com/wearetudublin">TU Dublin Twitter</a>
            <p>Click here and check our Facebook</p><a href="https://www.fac
ebook.com/TechnologicalUniversityDublin.TUDublin/">TU Dublin Facebook</a>
            <p>Click here and check our webpage </p><a href="https://www.dit
.ie">TU Dublin </a>
          </section>
      </body>
    </html>
```

Define Document Object Model and draw a diagram that represents the DOM tree for the code example above.

(10 marks)


The Document Object Model (DOM) is a programming interface for HTML and XML documents that represents the structure of a document as a tree of objects. Each object in the tree represents a part of the document, such as an element, an attribute, or a piece of text. The DOM allows developers to manipulate the content, structure, and style of a document dynamically, using a programming language such as JavaScript.

Here is a DOM tree diagram representing the code example you provided:

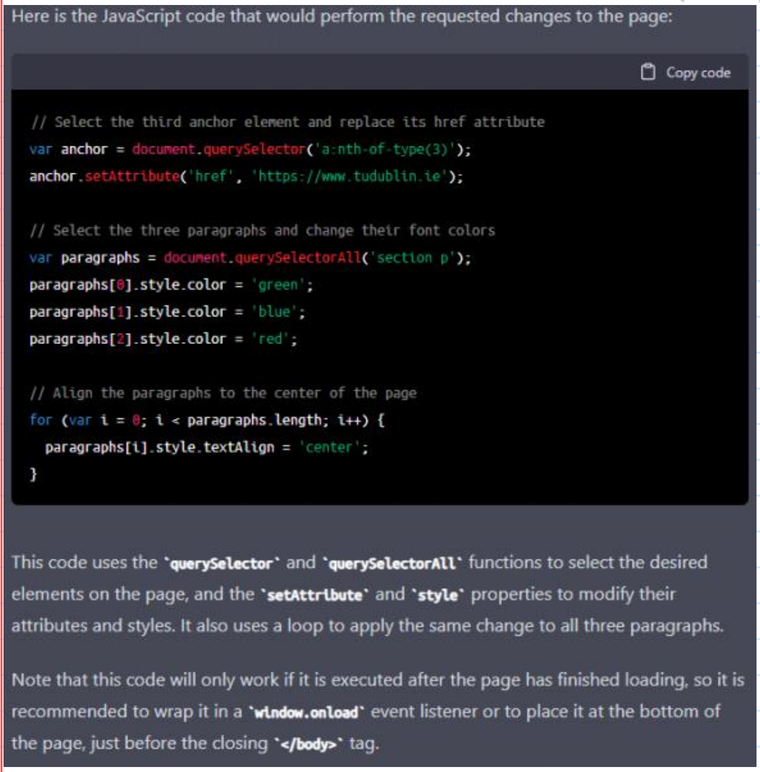
```
html
  head
    title (Technological University Dublin)
  body
    section
      img (logo.png)
      p
        a (TU Dublin Twitter)
      p
        a (TU Dublin Facebook)
      p
        a (TU Dublin)
```

(c) Consider you were asked to develop a Chrome extension to modify the page described in question 1b. This extension should:

- Replace the href content inside the third anchor element by https://www.tudublin.ie.
- Manipulate the style of the section elements by changing the font colour of the content in the first paragraph to green, in the second paragraph to blue, and in the third paragraph to red.
- Align the first three paragraphs in the centre of the page.

Provide the JavaScript code that can perform those changes. You should use pure JavaScript and must not use libraries.

(10 marks)

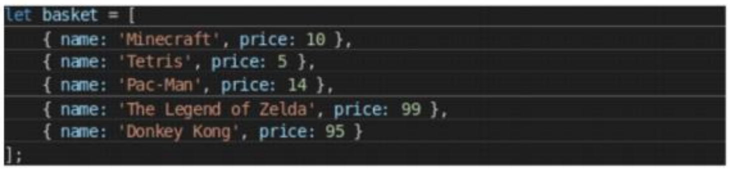

Here is the JavaScript code that would perform the requested changes to the page:

```javascript
// Select the third anchor element and replace its href attribute
var anchor = document.querySelector('a:nth-of-type(3)');
anchor.setAttribute('href', 'https://www.tudublin.ie');

// Select the three paragraphs and change their font colors
var paragraphs = document.querySelectorAll('section p');
paragraphs[0].style.color = 'green';
paragraphs[1].style.color = 'blue';
paragraphs[2].style.color = 'red';

// Align the paragraphs to the center of the page
for (var i = 0; i < paragraphs.length; i++) {
  paragraphs[i].style.textAlign = 'center';
}
```

This code uses the `querySelector` and `querySelectorAll` functions to select the desired elements on the page, and the `setAttribute` and `style` properties to modify their attributes and styles. It also uses a loop to apply the same change to all three paragraphs.

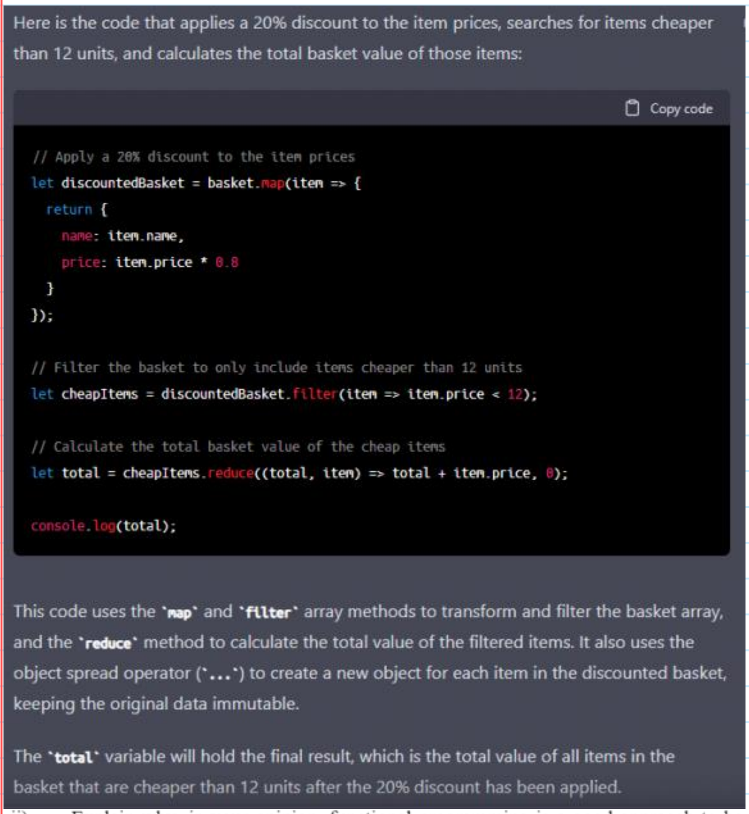
Note that this code will only work if it is executed after the page has finished loading, so it is recommended to wrap it in a `window.onload` event listener or to place it at the bottom of the page, just before the closing `</body>` tag.

(d) Functional programming is the process of building software by composing pure functions, avoiding shared state, mutable data, and side-effects. Check the following code:



```
let basket = [
    { name: 'Minecraft', price: 10 },
    { name: 'Tetris', price: 5 },
    { name: 'Pac-Man', price: 14 },
    { name: 'The Legend of Zelda', price: 99 },
    { name: 'Donkey Kong', price: 95 }
];
```

i) Using the functional programming property of keeping the data immutable, write a code that applies a 20% discount coupon code on the item prices, search for those cheaper than 12 units, and calculate the total basket value if you bought all the products cheaper than 12 units.



Here is the code that applies a 20% discount to the item prices, searches for items cheaper than 12 units, and calculates the total basket value of those items:

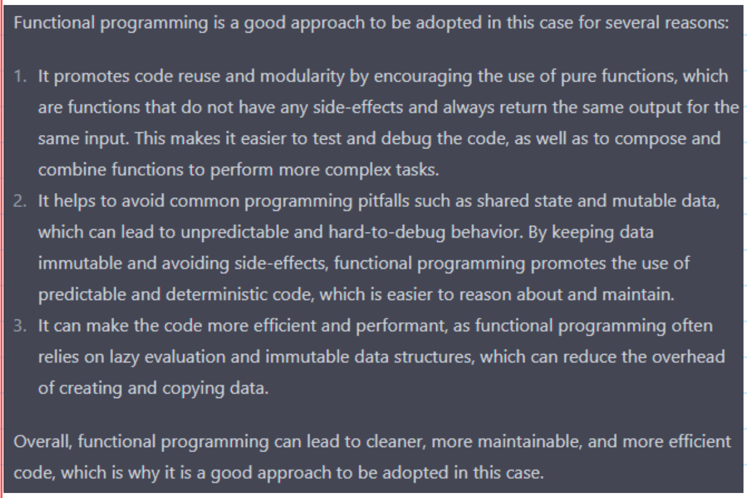
```
// Apply a 20% discount to the item prices
let discountedBasket = basket.map(item => {
  return {
    name: item.name,
    price: item.price * 0.8
  }
});

// Filter the basket to only include items cheaper than 12 units
let cheapItems = discountedBasket.filter(item => item.price < 12);

// Calculate the total basket value of the cheap items
let total = cheapItems.reduce((total, item) => total + item.price, 0);

console.log(total);
```

This code uses the `map` and `filter` array methods to transform and filter the basket array, and the `reduce` method to calculate the total value of the filtered items. It also uses the object spread operator (`...`) to create a new object for each item in the discounted basket, keeping the original data immutable.

The `total` variable will hold the final result, which is the total value of all items in the basket that are cheaper than 12 units after the 20% discount has been applied.

ii) Explain why, in your opinion, functional programming is a good approach to be adopted in this case.

(18 marks)

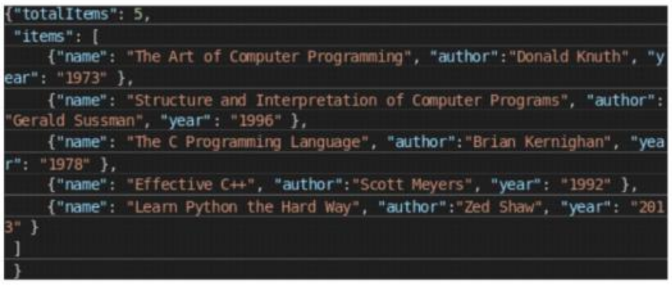


Functional programming is a good approach to be adopted in this case for several reasons:

1. It promotes code reuse and modularity by encouraging the use of pure functions, which are functions that do not have any side-effects and always return the same output for the same input. This makes it easier to test and debug the code, as well as to compose and combine functions to perform more complex tasks.
2. It helps to avoid common programming pitfalls such as shared state and mutable data, which can lead to unpredictable and hard-to-debug behavior. By keeping data immutable and avoiding side-effects, functional programming promotes the use of predictable and deterministic code, which is easier to reason about and maintain.
3. It can make the code more efficient and performant, as functional programming often relies on lazy evaluation and immutable data structures, which can reduce the overhead of creating and copying data.

Overall, functional programming can lead to cleaner, more maintainable, and more efficient code, which is why it is a good approach to be adopted in this case.

(2) (a) Consider the following JSON file:



```
{"totalItems": 5,
 "items": [
     {"name": "The Art of Computer Programming", "author":"Donald Knuth", "y
ear": "1973" },
     {"name": "Structure and Interpretation of Computer Programs", "author":
"Gerald Sussman", "year": "1996" },
     {"name": "The C Programming Language", "author":"Brian Kernighan", "yea
r": "1978" },
     {"name": "Effective C++", "author":"Scott Meyers", "year": "1992" },
     {"name": "Learn Python the Hard Way", "author":"Zed Shaw", "year": "201
3" }
 ]
}
```

Explain, with your own words, the main differences between JSON and XML, and the advantages and disadvantages of using each of them. Using the JavaScript Fetch API, write a code to retrieve the data from the JSON example above.

(12 marks)

JSON (JavaScript Object Notation) and XML (eXtensible Markup Language) are both widely used data interchange formats that are used to transmit structured data over the internet or between applications.

One main difference between JSON and XML is the syntax: JSON uses a simpler and more compact syntax, based on the syntax of JavaScript objects, while XML uses a more verbose and flexible syntax, based on tags and attributes. JSON is generally easier to read and write for humans, while XML can be more expressive and support more advanced features such as namespaces and validation.

Another difference is the data types that each format supports: JSON supports a limited set of data types, including strings, numbers, booleans, arrays, and objects, while XML can represent any data type using a combination of tags and attributes.

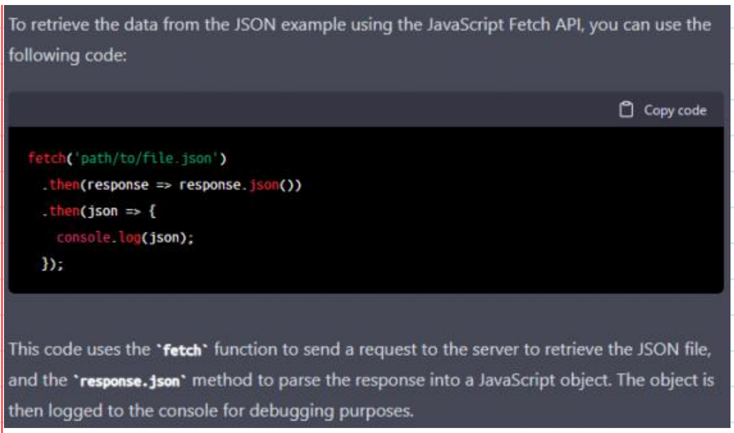
Here are some advantages and disadvantages of using each format:

JSON:

- Advantages:
  - Simplicity and lightweight: JSON is easier to read and write, and it produces smaller payloads compared to XML.
  - Efficient parsing: JSON can be easily parsed by modern browsers and programming languages, which makes it a good choice for data transfer between client and server.
  - Better support in JavaScript: JSON is natively supported in JavaScript, which makes it easier to work with in this language.
- Disadvantages:
  - Limited expressiveness: JSON does not support advanced features such as namespaces and validation, which can be a limitation for more complex data structures.
  - Lack of support for non-text data: JSON does not have a standard way of representing binary data, which can make it less suitable for transferring non-textual data such as images or audio files.

XML:

- Advantages:
  - Flexibility and expressiveness: XML supports a wide range of data types and has a rich set of features such as namespaces, validation, and data binding, which makes it suitable for representing complex data structures.
  - Wide adoption: XML is widely used in a variety of applications and industries, which makes it a good choice for data interoperability.
  - Good support for non-text data: XML has several mechanisms for representing and encoding non-text data, such as Base64 encoding and CDATA sections.
- Disadvantages:
  - Verbosity: XML is more verbose and less compact compared to JSON, which can make it less efficient for data transfer.
  - Complex parsing: XML has a more complex syntax and requires more processing to parse and validate compared to JSON, which can affect performance.
  - Lack of native support in some languages: Some programming languages, such as JavaScript, do not have native support for XML, which can make it more difficult to work with.

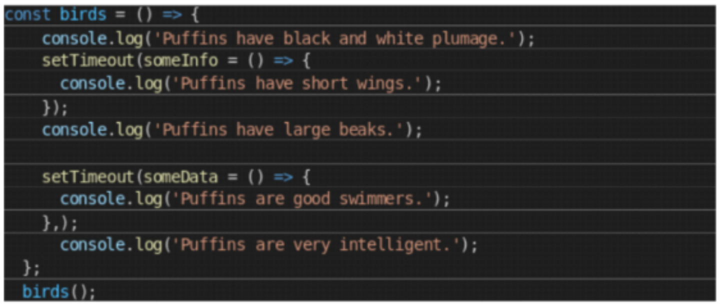


To retrieve the data from the JSON example using the JavaScript Fetch API, you can use the following code:

```
fetch('path/to/file.json')
  .then(response => response.json())
  .then(json => {
    console.log(json);
  });
```

This code uses the `fetch` function to send a request to the server to retrieve the JSON file, and the `response.json` method to parse the response into a JavaScript object. The object is then logged to the console for debugging purposes.

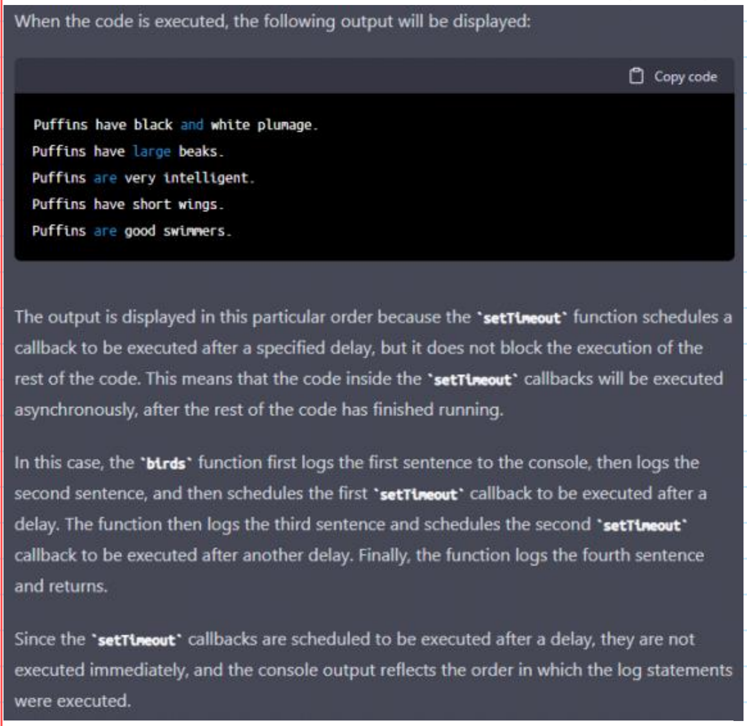
(b) Consider the following code:



```
const birds = () => {
    console.log('Puffins have black and white plumage.');
    setTimeout(someInfo = () => {
      console.log('Puffins have short wings.');
    });
    console.log('Puffins have large beaks.');

    setTimeout(someData = () => {
      console.log('Puffins are good swimmers.');
    },);
      console.log('Puffins are very intelligent.');
  };
  birds();
```

Describe what will be the output (in the right order) when the code is executed. Explain why the output is displayed in this particular order.

(18 Marks)



When the code is executed, the following output will be displayed:

```
Puffins have black and white plumage.
Puffins have large beaks.
Puffins are very intelligent.
Puffins have short wings.
Puffins are good swimmers.
```

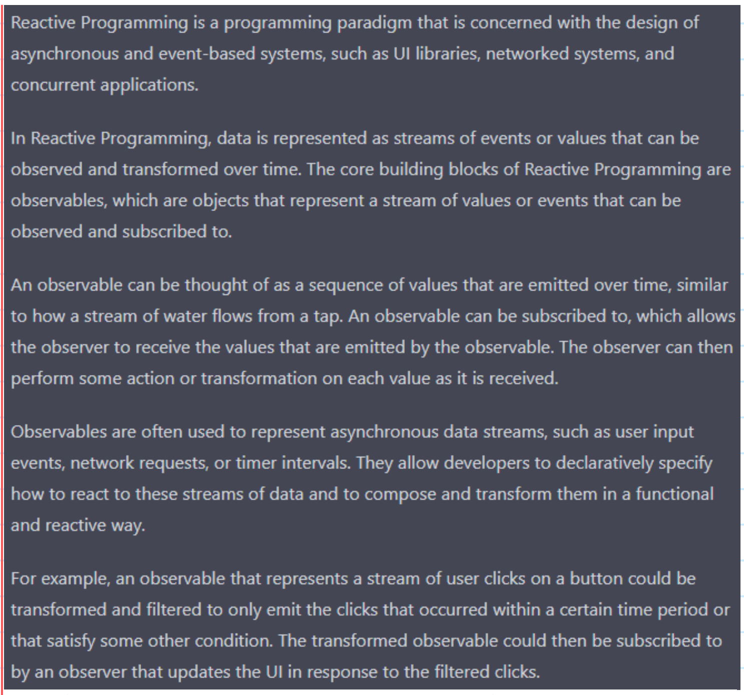
The output is displayed in this particular order because the `setTimeout` function schedules a callback to be executed after a specified delay, but it does not block the execution of the rest of the code. This means that the code inside the `setTimeout` callbacks will be executed asynchronously, after the rest of the code has finished running.

In this case, the `birds` function first logs the first sentence to the console, then logs the second sentence, and then schedules the first `setTimeout` callback to be executed after a delay. The function then logs the third sentence and schedules the second `setTimeout` callback to be executed after another delay. Finally, the function logs the fourth sentence and returns.

Since the `setTimeout` callbacks are scheduled to be executed after a delay, they are not executed immediately, and the console output reflects the order in which the log statements were executed.

(3) (a) Describe Reactive Programming and define what is an observable in Reactive Programming.

(6 marks)



Reactive Programming is a programming paradigm that is concerned with the design of asynchronous and event-based systems, such as UI libraries, networked systems, and concurrent applications.

In Reactive Programming, data is represented as streams of events or values that can be observed and transformed over time. The core building blocks of Reactive Programming are observables, which are objects that represent a stream of values or events that can be observed and subscribed to.

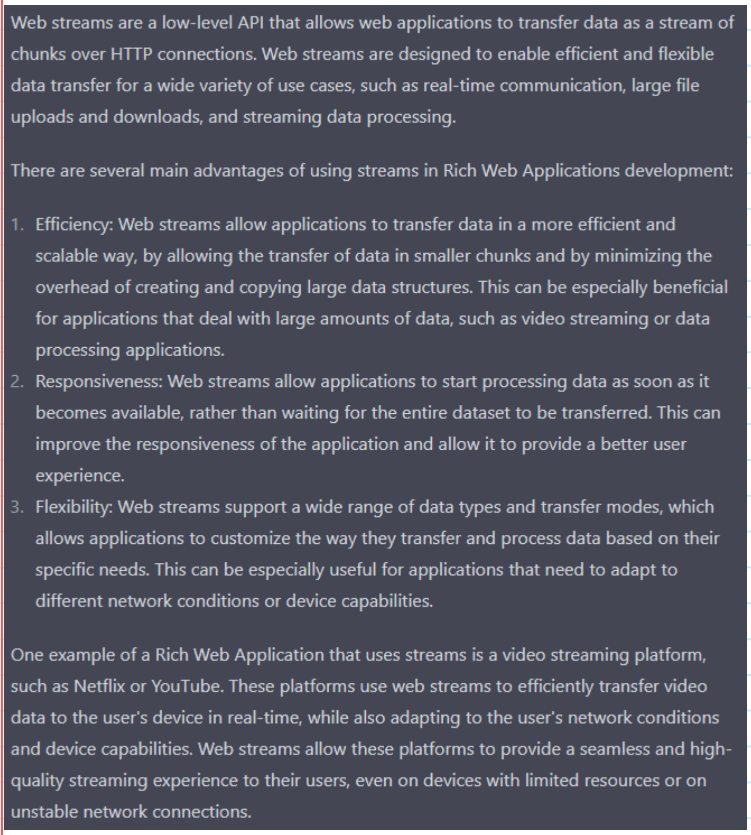
An observable can be thought of as a sequence of values that are emitted over time, similar to how a stream of water flows from a tap. An observable can be subscribed to, which allows the observer to receive the values that are emitted by the observable. The observer can then perform some action or transformation on each value as it is received.

Observables are often used to represent asynchronous data streams, such as user input events, network requests, or timer intervals. They allow developers to declaratively specify how to react to these streams of data and to compose and transform them in a functional and reactive way.

For example, an observable that represents a stream of user clicks on a button could be transformed and filtered to only emit the clicks that occurred within a certain time period or that satisfy some other condition. The transformed observable could then be subscribed to by an observer that updates the UI in response to the filtered clicks.

(b) Define web streams and explain the main advantages of using streams in Rich Web Applications development. Thinking about real-life applications, exemplify one rich apps that use or can benefit of using streams.

(6 marks)



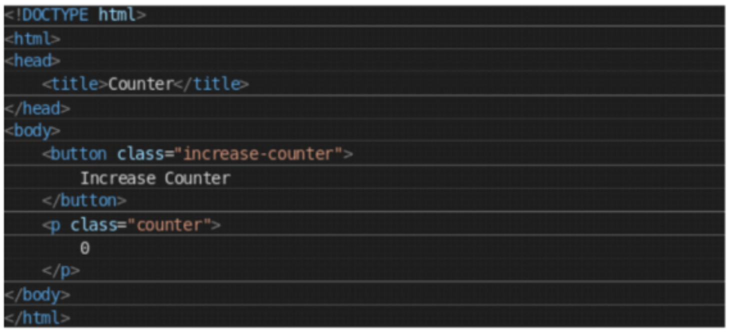
Web streams are a low-level API that allows web applications to transfer data as a stream of chunks over HTTP connections. Web streams are designed to enable efficient and flexible data transfer for a wide variety of use cases, such as real-time communication, large file uploads and downloads, and streaming data processing.

There are several main advantages of using streams in Rich Web Applications development:

1. Efficiency: Web streams allow applications to transfer data in a more efficient and scalable way, by allowing the transfer of data in smaller chunks and by minimizing the overhead of creating and copying large data structures. This can be especially beneficial for applications that deal with large amounts of data, such as video streaming or data processing applications.
2. Responsiveness: Web streams allow applications to start processing data as soon as it becomes available, rather than waiting for the entire dataset to be transferred. This can improve the responsiveness of the application and allow it to provide a better user experience.
3. Flexibility: Web streams support a wide range of data types and transfer modes, which allows applications to customize the way they transfer and process data based on their specific needs. This can be especially useful for applications that need to adapt to different network conditions or device capabilities.

One example of a Rich Web Application that uses streams is a video streaming platform, such as Netflix or YouTube. These platforms use web streams to efficiently transfer video data to the user's device in real-time, while also adapting to the user's network conditions and device capabilities. Web streams allow these platforms to provide a seamless and high-quality streaming experience to their users, even on devices with limited resources or on unstable network connections.

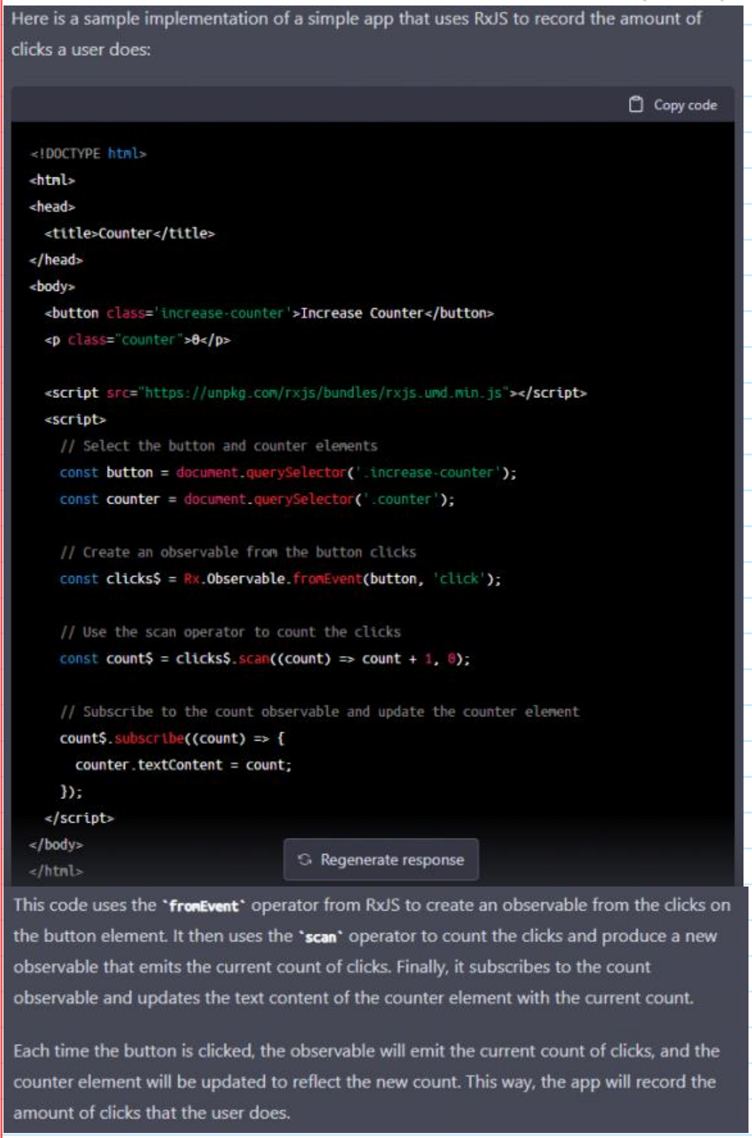
(c) Consider the following code:



```
<!DOCTYPE html>
<html>
<head>
    <title>Counter</title>
</head>
<body>
    <button class="increase-counter">
        Increase Counter
    </button>
    <p class="counter">
        0
    </p>
</body>
</html>
```

Using RxJS, implement a simple app that records the amount of clicks a user does.

(10 Marks)



Here is a sample implementation of a simple app that uses RxJS to record the amount of clicks a user does:

```
<!DOCTYPE html>
<html>
<head>
  <title>Counter</title>
</head>
<body>
  <button class='increase-counter'>Increase Counter</button>
  <p class="counter">0</p>

  <script src="https://unpkg.com/rxjs/bundles/rxjs.umd.min.js"></script>
  <script>
    // Select the button and counter elements
    const button = document.querySelector('.increase-counter');
    const counter = document.querySelector('.counter');

    // Create an observable from the button clicks
    const clicks$ = Rx.Observable.fromEvent(button, 'click');

    // Use the scan operator to count the clicks
    const count$ = clicks$.scan((count) => count + 1, 0);

    // Subscribe to the count observable and update the counter element
    count$.subscribe((count) => {
      counter.textContent = count;
    });
  </script>
</body>
</html>
```

This code uses the `fromEvent` operator from RxJS to create an observable from the clicks on the button element. It then uses the `scan` operator to count the clicks and produce a new observable that emits the current count of clicks. Finally, it subscribes to the count observable and updates the text content of the counter element with the current count.

Each time the button is clicked, the observable will emit the current count of clicks, and the counter element will be updated to reflect the new count. This way, the app will record the amount of clicks that the user does.