DWA_03.4 Knowledge Check_DWA3.1

---

1. Please show how you applied a Markdown File to a piece of your code.

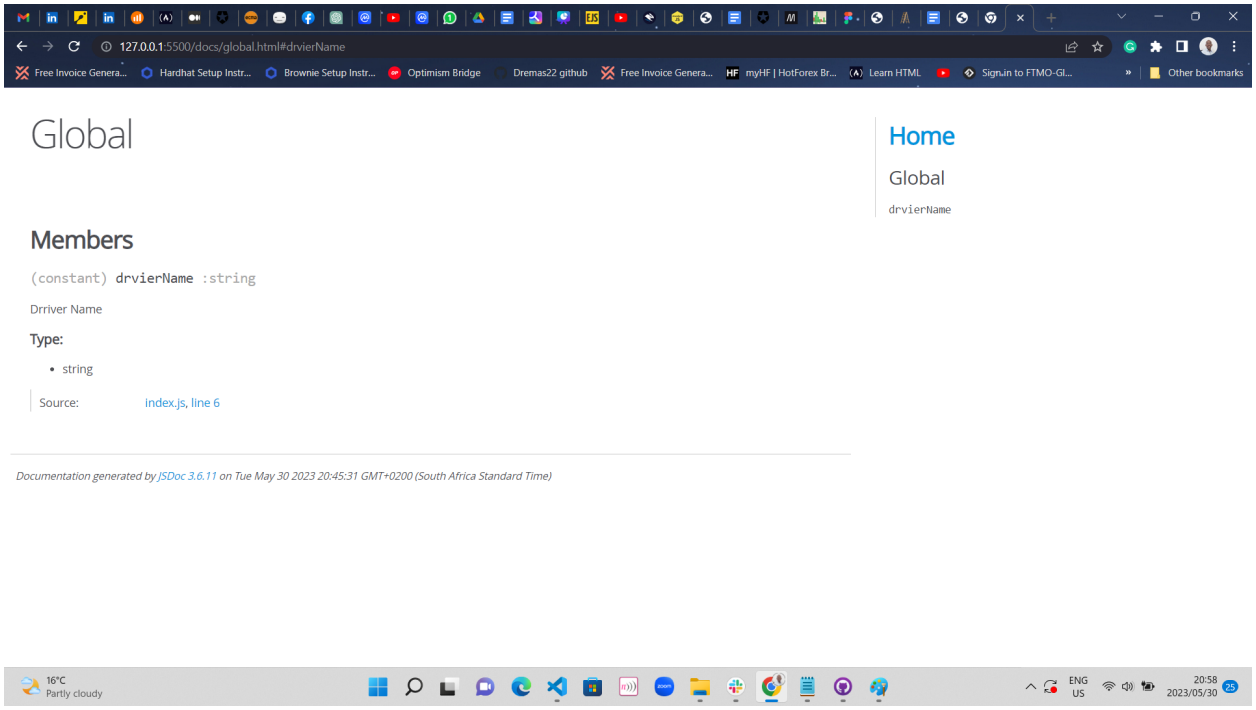
```
# Maxi taxi App

Our Maxi Taxi App aims to enhance the transportation ecosystem, bringing convenience, efficiency, and reliability to both drivers and commuters. Experience a seamless ride-hailing experience, eliminate the need for hand signals, and effortlessly connect with nearby maxi taxis.

- [Maxi taxi App](#maxi-taxi-app)
  - [Features:](#features)
  - [Requirements:](#requirements)
- [Getting Started:](#getting-started)


## Features:

- ❤️ User firendly
- 💪 Reliable
- 💰 Productive

## Requirements:

- An IDE platform [Visual Studio Code](https://code.visualstudio.com/).
- Basic [HTML, CSS and JavaScript](https://developer.mozilla.org/en-US/docs/Learn)
- You can use browser like [Chrome Browser](https://www.google.com/chrome)

# Getting Started:

```
```
```

```
1. Clone the repository: git Clone

2. Run a localhost server and open
DWA3-1.md

3. Open - [Maxi taxi App](#maxi-taxi-app)
  - [Features:](#features)
  - [Requirements:](#requirements)
- [Getting Started:](#getting-started)
```

---

2. Please show how you applied JSDoc Comments to a piece of your code.

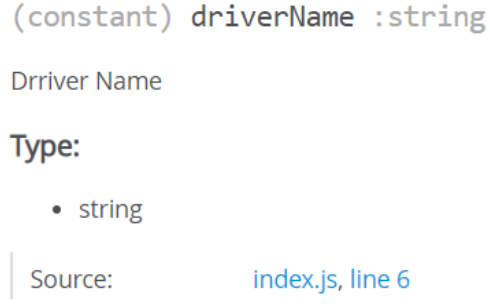
```
/**
 * Drriver Name
 * @type {string}
 */

const drvierName = "Tshepo Masilo";
```



---

3. Please show how you applied the @ts-check annotation to a piece of your code.

```
//  @ts-check

/**
 * Array of Maxi taxis registered on the app
 * @param {Array} arrayMaxiTaxis
 */
const arrayMaxiTaxis = [maxi1, maxi2, maxi3]
```



---

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.

```
/**
 * Compound interst module
 * @module compound interest formula
 */

/**
 * Calculates the future value using the compound interest formula.
 * @param {number} principal - The principal amount (initial
investment).
 * @param {number} interestRate - The annual interest rate (expressed
as a decimal).
 * @param {number} compoundingPeriods - The number of compounding
periods per year.
 * @param {number} years - The duration of the investment in years.
 * @returns {number} The future value after interest.
 */
function calculateCompoundInterest(principal, interestRate,
compoundingPeriods, years) {
    const n = compoundingPeriods;
    const t = years;
    const A = principal * Math.pow(1 + interestRate / n, n * t);
    return A;
  }

  // Example usage
  const principalAmount = 1000; // $1000 initial investment
  const annualInterestRate = 0.05; // 5% annual interest rate
  const compoundingPeriodsPerYear = 12; // Compounded monthly
  const investmentDurationYears = 5; // 5 years

  const futureValue = calculateCompoundInterest(
```

```
    principalAmount,
    annualInterestRate,
    compoundingPeriodsPerYear,
    investmentDurationYears
);
```

Module: compound interest formula

Compound interst module

Source: compound-interest-formula.js, line 1

Methods

(inner) calculateCompoundInterest(principal, interestRate, compoundingPeriods, years) → {number}

Calculates the future value using the compound interest formula.

Parameters:

Name	Type	Description
principal	number	The principal amount (initial investment).
interestRate	number	The annual interest rate (expressed as a decimal).
compoundingPeriods	number	The number of compounding periods per year.