** - Question 1: Explain how FastAPI handles asynchronous requests and its benefits over synchronous code in Python.**

FastAPI also supports asynchronous programming, which allows for better scalability and responsiveness. It uses the async and await keywords to handle asynchronous operations, such as making HTTP requests or querying databases, without blocking the execution of other tasks. This makes it ideal for building highly concurrent applications that can handle a large number of requests.

** - Question 2: Describe how dependency injection works in FastAPI and give an example of its practical use.**

[FastAPI](https://www.linkedin.com/company/fastapi/) has a built-in dependency injection system that makes it easy to inject dependencies into your code. To use dependency injection in FastAPI, you can use the **Depends()** dependency.

The **Depends()** dependency is a function that takes a dependency as its argument and returns the dependency

** - Question 3: Code walkthrough**

* Import FastAPI.
* Create an app instance.
* Write a **path operation decorator** (like @app.get("/")).
* Write a **path operation function** (like def root(): ... above).
* Run the development server (like uvicorn main:app --reload).