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

Ans: In Fast API, it handles asynchronous requests using async and awaits keywords in Python by leveraging the asyncio package. Asynchronous functions handle the request concurrently without waiting to complete the current task; they can handle another request in the meantime.

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

Ans: Dependency injection allows you to declare dependencies for your API endpoints, making your code modular, reusable, and easier to test. FastAPI's dependency injection system is based on Python type hints.

Example:

from fastapi import FastAPI, Depends

app = FastAPI()

def get\_db\_connection():

# For database connection

return db\_connection

def get\_current\_user():

# Code to get the current user

return current\_user

@app.get("/items/")

async def read\_items(db: DBSession = Depends(get\_db\_connection), user: User = Depends(get\_current\_user)):

Question 3: Code walkthrough

Ans: A code walkthrough is a piece of code to analyze and find the core logic behind it to understand. It is helpful when you need to add a feature or modify an existing feature in the code.