FastAPI

Date: - 31-10-2025

Day: - 1

Topics Covered: -

1. Introduction to FastAPI
2. Installing FastAPI and Uvicorn
3. What is an API (Concept Explanation)
4. JSON Explanation and Response Format
5. Creating Multiple Endpoints (GET Method)
6. Path / Endpoint Parameters
7. Query Parameters
8. Request Body & POST Method
9. PUT Method
10. DELETE Method

PROBLEM: -

GET Method

```
from fastapi import FastAPI
app = FastAPI()
@app.get("/")
def home():
return {"message": "Welcome to FastAPI Practice!"}
```

Path Parameter

```
if item_id in inventory:
  return inventory[item_id]
```

```
    return {"error": "Item not found"} from fastapi import FastAPI

app = FastAPI()

inventory = {
    1: {"name": "milk", "price": 60, "brand": "Amul"},
    2: {"name": "bread", "price": 40, "brand": "Britannia"},
    3: {"name": "butter", "price": 120, "brand": "Amul"}
}

@app.get("/get-item/{item_id}")
def get_item(item_id: int):
```

Query Parameters

```
from fastapi import FastAPI

app = FastAPI()

inventory = {
    1: {"name": "milk", "price": 60, "brand": "Amul"},
    2: {"name": "bread", "price": 40, "brand": "Britannia"},
    3: {"name": "butter", "price": 120, "brand": "Amul"}
}

@app.get("/search-by-brand/")
def search_items(brand: str):
    results = [item for item in inventory.values() if item["brand"].lower() == 
brand.lower()]
```

```
    if results:
        return {"brand": brand, "items": results}
    return {"message": f"No items found for brand '{brand}'"}
```

POST Method

```
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

class Item(BaseModel):
    name: str
    price: float
    brand: str

inventory = {}

@app.post("/add-item/{item_id}")
def add_item(item_id: int, item: Item):
    if item_id in inventory:
        return {"error": "Item already exists"}
    inventory[item_id] = item.dict()
    return {"message": "Item added successfully", "data": inventory[item_id]}
```

PUT Method

```
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()
```

```
class Item(BaseModel):
    name: str = None
    price: float = None
    brand: str = None


inventory = {
    1: {"name": "milk", "price": 60, "brand": "Amul"},
    2: {"name": "bread", "price": 40, "brand": "Britannia"}
}


@app.put("/update-item/{item_id}")
def update_item(item_id: int, item: Item):
    if item_id not in inventory:
        return {"error": "Item not found"}


    if item.name is not None:
        inventory[item_id]["name"] = item.name
    if item.price is not None:
        inventory[item_id]["price"] = item.price
    if item.brand is not None:
        inventory[item_id]["brand"] = item.brand


    return {"message": "Item updated successfully", "data": inventory[item_id]}
```

DELETE Method

```
from fastapi import FastAPI
```

```
app = FastAPI()


inventory = {
    1: {"name": "milk", "price": 60, "brand": "Amul"},
    2: {"name": "bread", "price": 40, "brand": "Britannia"},
    3: {"name": "butter", "price": 120, "brand": "Amul"}
}


@app.delete("/delete-item/{item_id}")
def delete_item(item_id: int):
    if item_id not in inventory:
        return {"error": "Item not found"}
    del inventory[item_id]
    return {"message": f"Item {item_id} deleted successfully"}
```