# **Section 1. Issues with the existing code:**

Existing code :

@app.route('/api/products', methods=['POST'])

def create\_product():

data = request.json

# Create new product

product = Product(

name=data['name'],

sku=data['sku'],

price=data['price'],

warehouse\_id=data['warehouse\_id']

)

db.session.add(product)

db.session.commit()

# Update inventory count

inventory = Inventory(

product\_id=product.id,

warehouse\_id=data['warehouse\_id'],

quantity=data['initial\_quantity']

)

db.session.add(inventory)

db.session.commit()

return {"message": "Product created", "product\_id": [product.id](http://product.id)}

# Issues:

1. **SKU uniqueness not enforced:**

***Problem*** : The code does not check whether the *sku* is already present in the database.

***Impact***: Duplicate SKUs could be created, leading to data inconsistency, inventory confusion, and inability to distinguish products correctly.

1. **Products can exist in multiple warehouses but inventory logic is flawed:**

***Problem***: The code assumes that when a new product is created, its inventory is only added for one warehouse.

***Impact***: If the product is later stocked in other warehouses, there is no mechanism to link multiple inventories to the same product.

1. **No validation of input data:**

***Problem***: The code directly accesses keys like data['name'] without checking if they exist or are valid.

***Impact***: If required fields are missing or malformed, it will throw an exception and fail silently or return a 500 error.

1. **Price is not validated for type or range:**

***Problem***: The code does not ensure price is a decimal and non-negative.

***Impact***: Wrong data types or negative prices could cause issues in reporting, billing, and calculations.

1. **Initial inventory count is assumed to always be provided:**

***Problem***: data['initial\_quantity'] is directly used without checking its presence or ensuring it's non-negative.

***Impact***: Missing or negative values could break the inventory logic.

1. **No transaction handling:**

***Problem***: The two db.session.commit() calls are separate.

***Impact***: If the second commit fails (e.g., invalid quantity), the product is still created but the inventory is missing → inconsistent state.

1. ***Missing error handling:***

***Problem***: The provided warehouse\_id is used without checking if the warehouse exists.

***Impact***: Products and inventory could be linked to invalid warehouses.