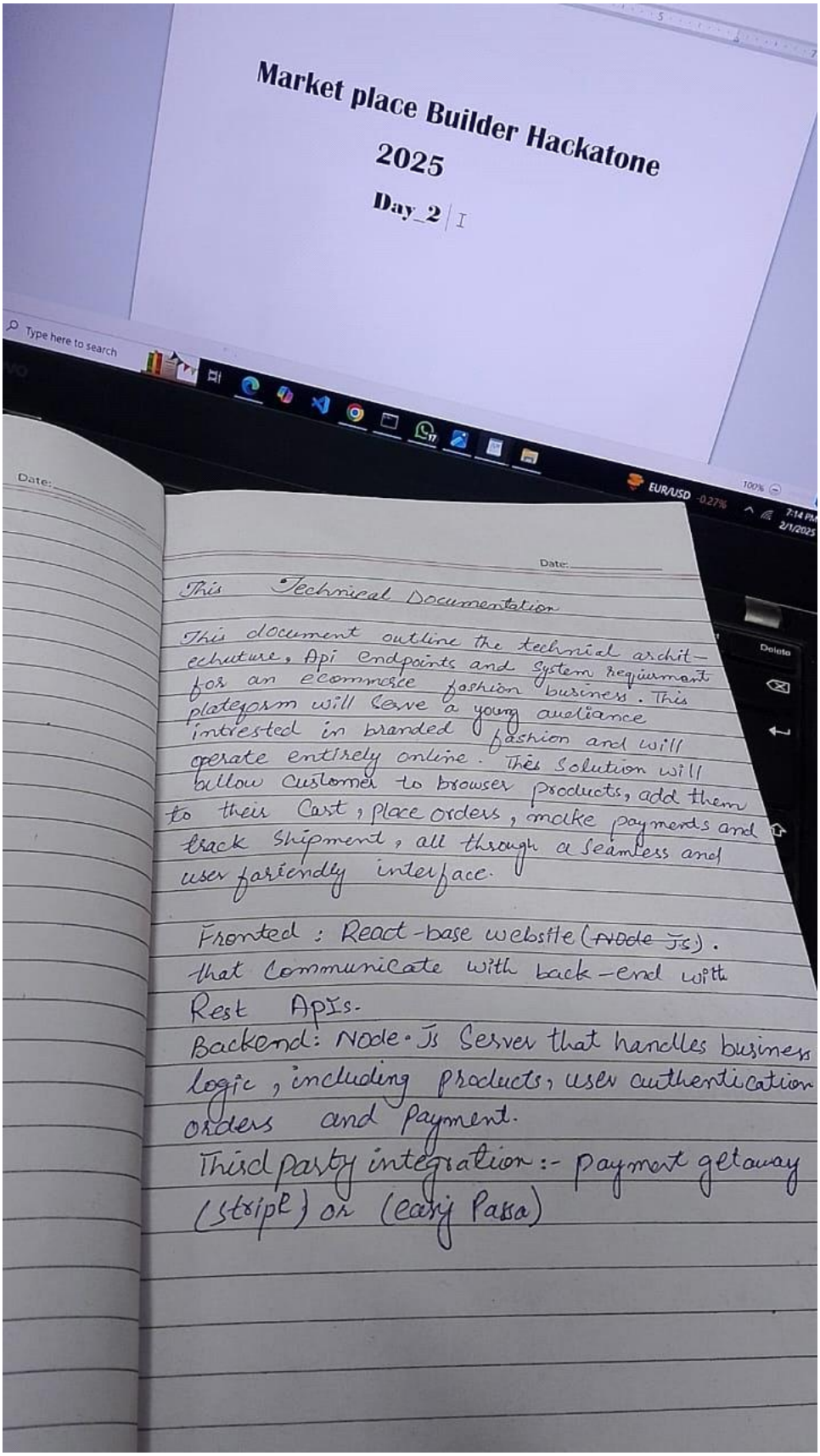
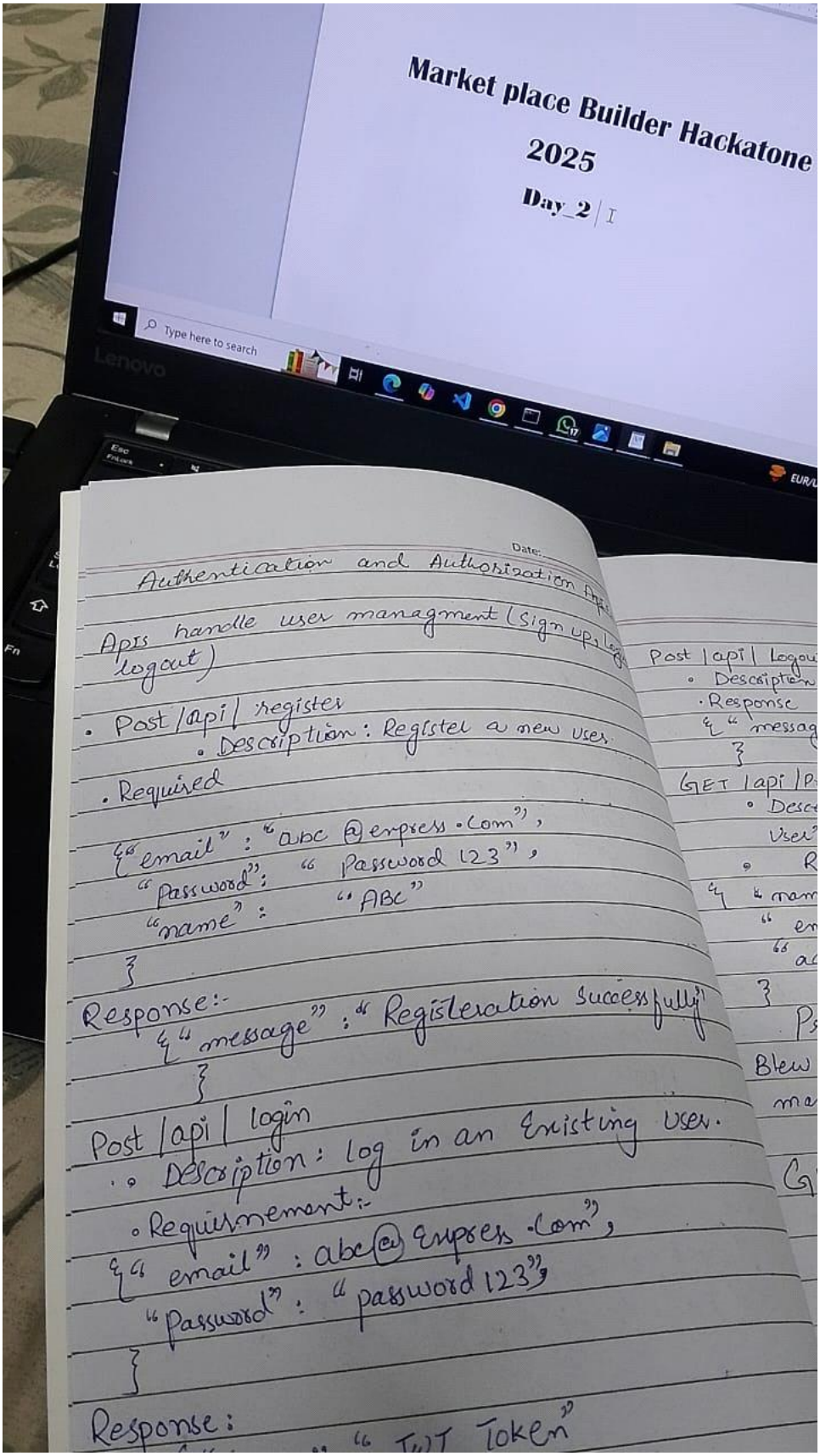
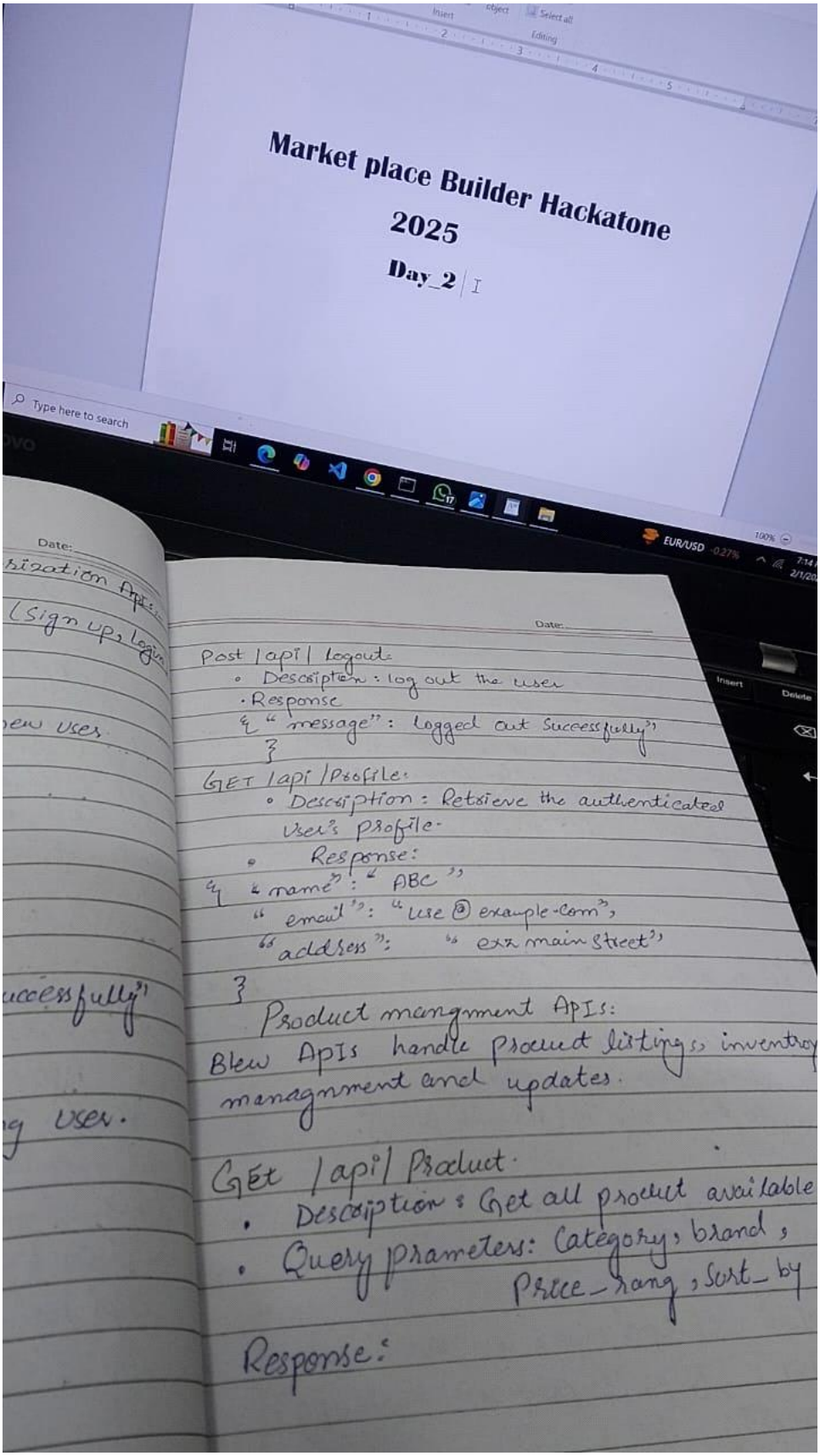
Market place Builder Hackatone

2025

Day_2


Market place Builder Hackatone
2025
Day_2
Type here to search
EUR/USD -0.27%
Date:
This Technical Documentation
This document outline the technical architecture, Api endpoints and system requirment for an ecommerce fashion business. This plateform will serve a young audiance intrested in branded fashion and will operate entirely online. This solution will allow customer to browser products, add them to their Cart, place orders, make payments and track shipment, all through a seamless and user faricndly interface.
Fronted : React-base website (Node Js).
that communicate with back-end with
Rest Apis.
Backend: Node.Js Server that handles business
logic, including products, user authentication
orders and payment.
Third party integration :- payment getaway
(stripe) or (easy Paisa)


Market place Builder Hackatone
2025
Day_2
Type here to search
Lenovo
Date:
Authentication and Authorization Apis
Apis handle user managment (Sign up, Login, logout)
• Post /api/ register
◦ Description: Register a new user.
• Required
{"email": "abc@erpress.com",
"Password": "Password123",
"name": "ABC"
}
Response:-
{"message": "Registeration Successfully"
}
Post /api/ login
◦ Description: Log in an Existing user.
◦ Requirnement:-
{"email": abc@erpress.com",
"Password": "password123"
}
Response:
"JWT Token"
Post /api/ Logou
◦ Descriptio
• Response
{"messag
}
GET /api/P
◦ Desc
User
◦ R
{"nam
"em
"ac
}
Blew
ma



Market place Builder Hackatone

2025

Day_2

Post /api/ logout

- Description: log out the user

- Response

{ "message": Logged out successfully"

}

GET /api/Profile:

- Description: Retsieve the authenticated User's Profile.
- Response:

{ "name": "ABC",

"email": "use@example.com",

"address": "ex main street"

}

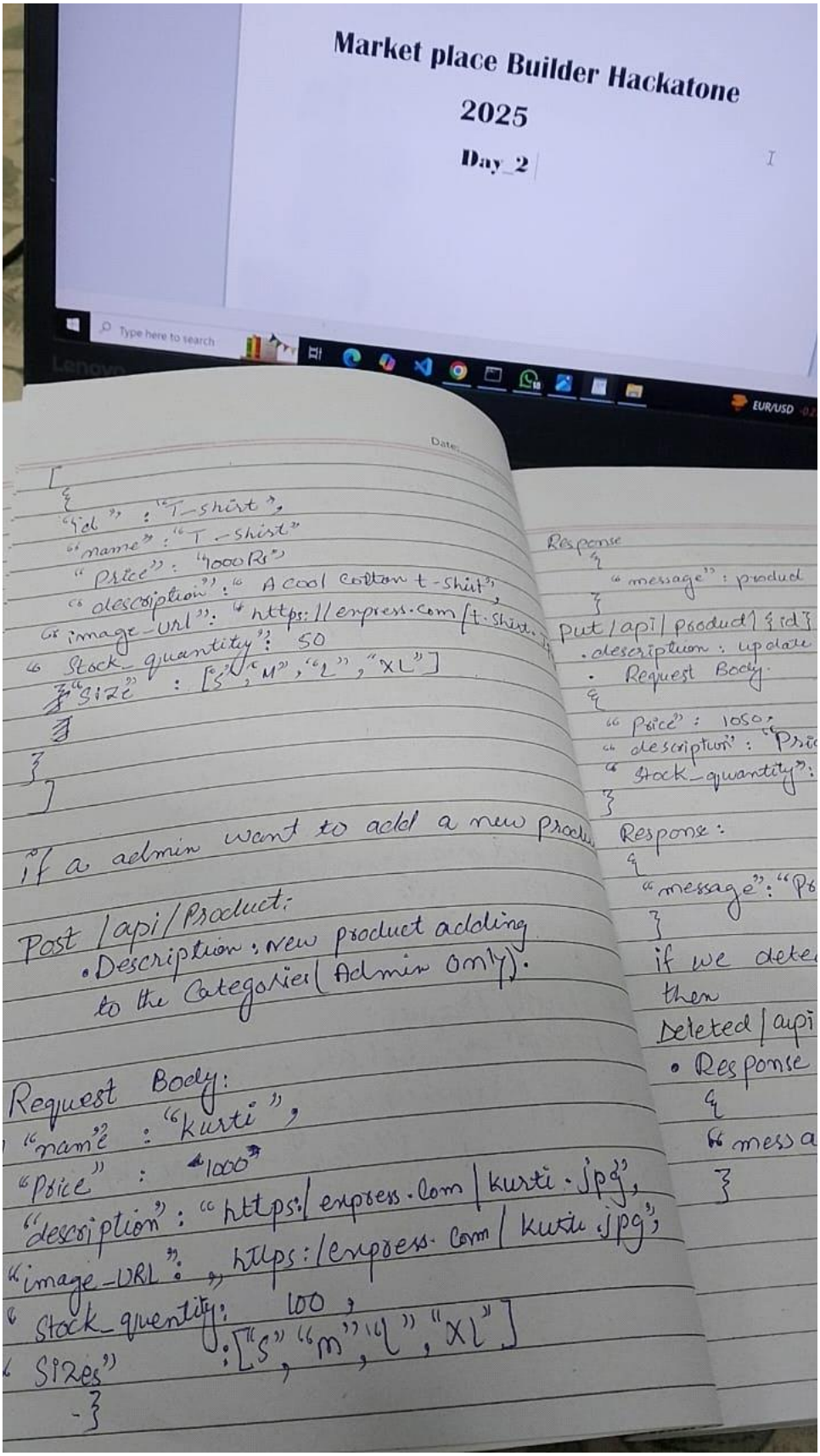
Product mangment APIs:

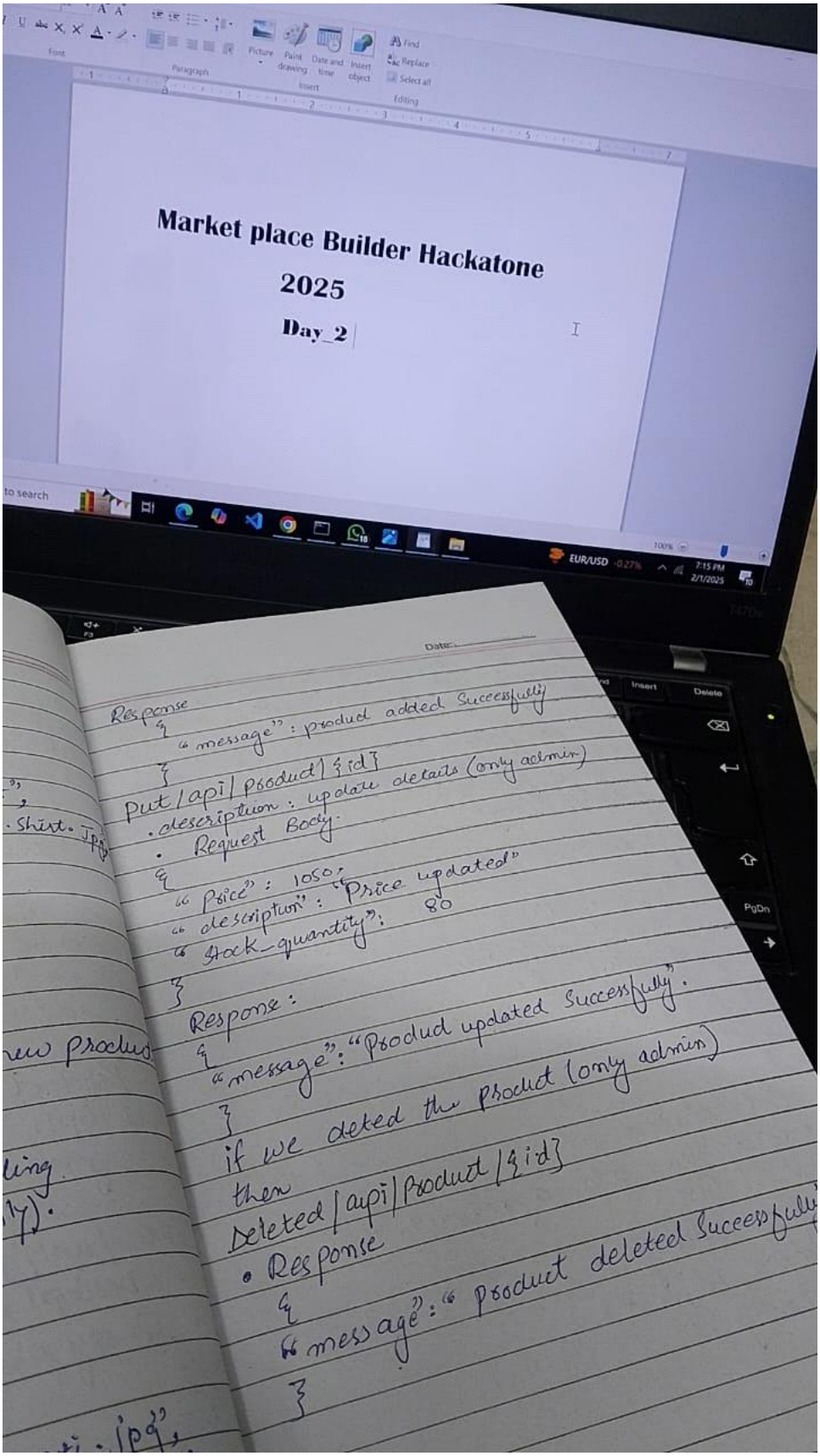
Blew APIs handle Product listings, inventroy managment and updates.

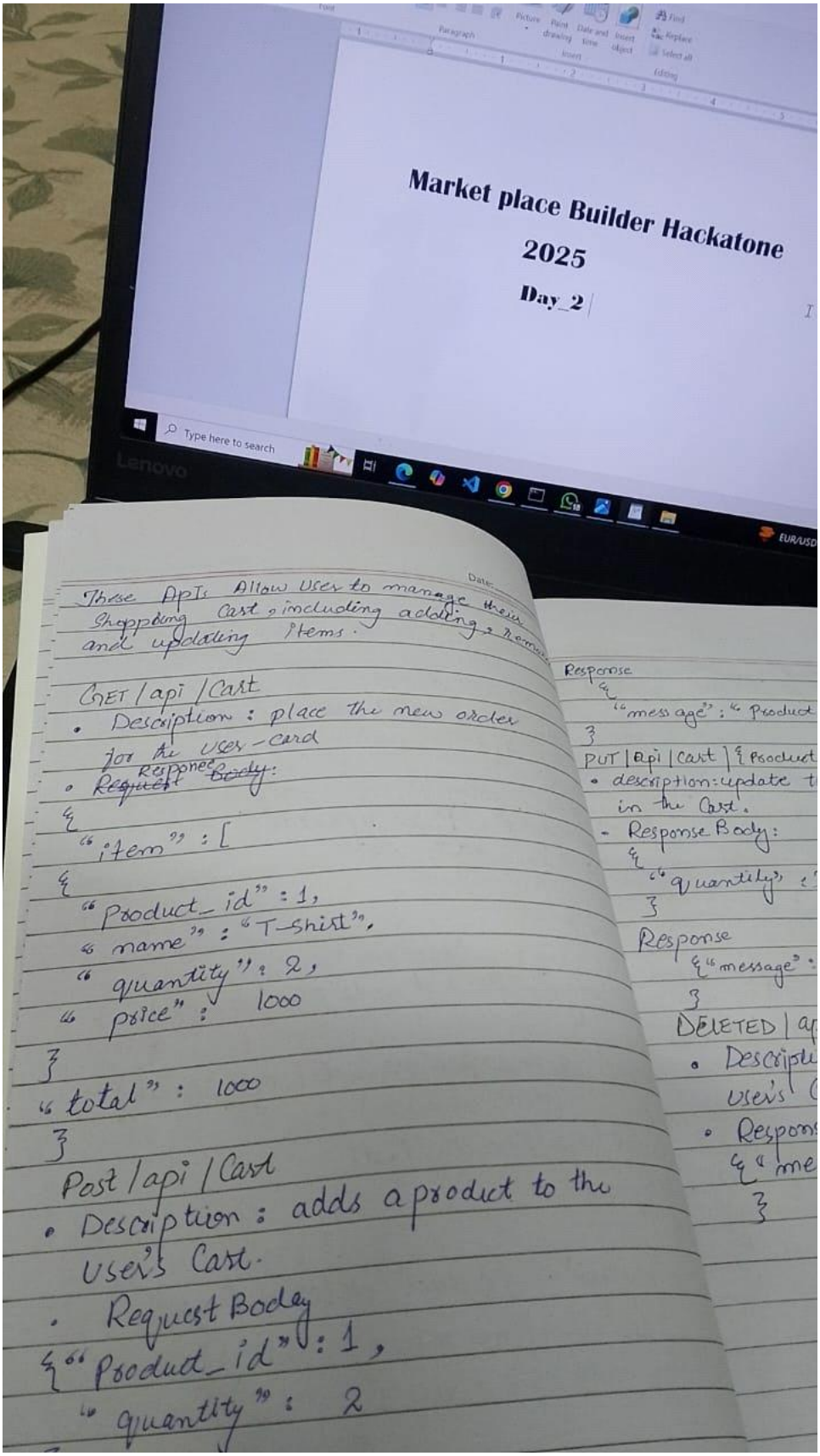
Get /api/ Product.

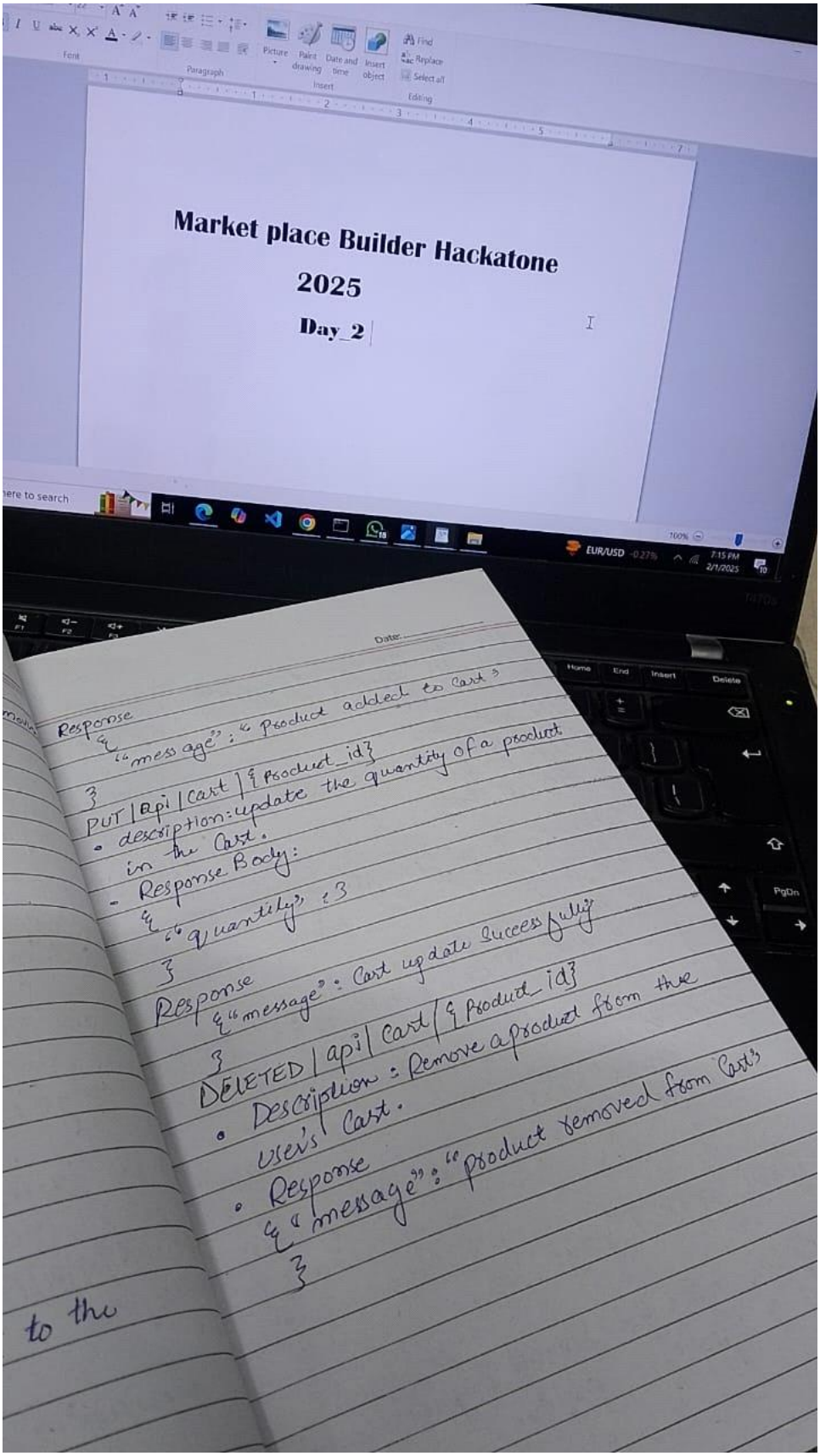
- Description: Get all product available
- Query prameters: Category, brand, Price_rang, Sort_by

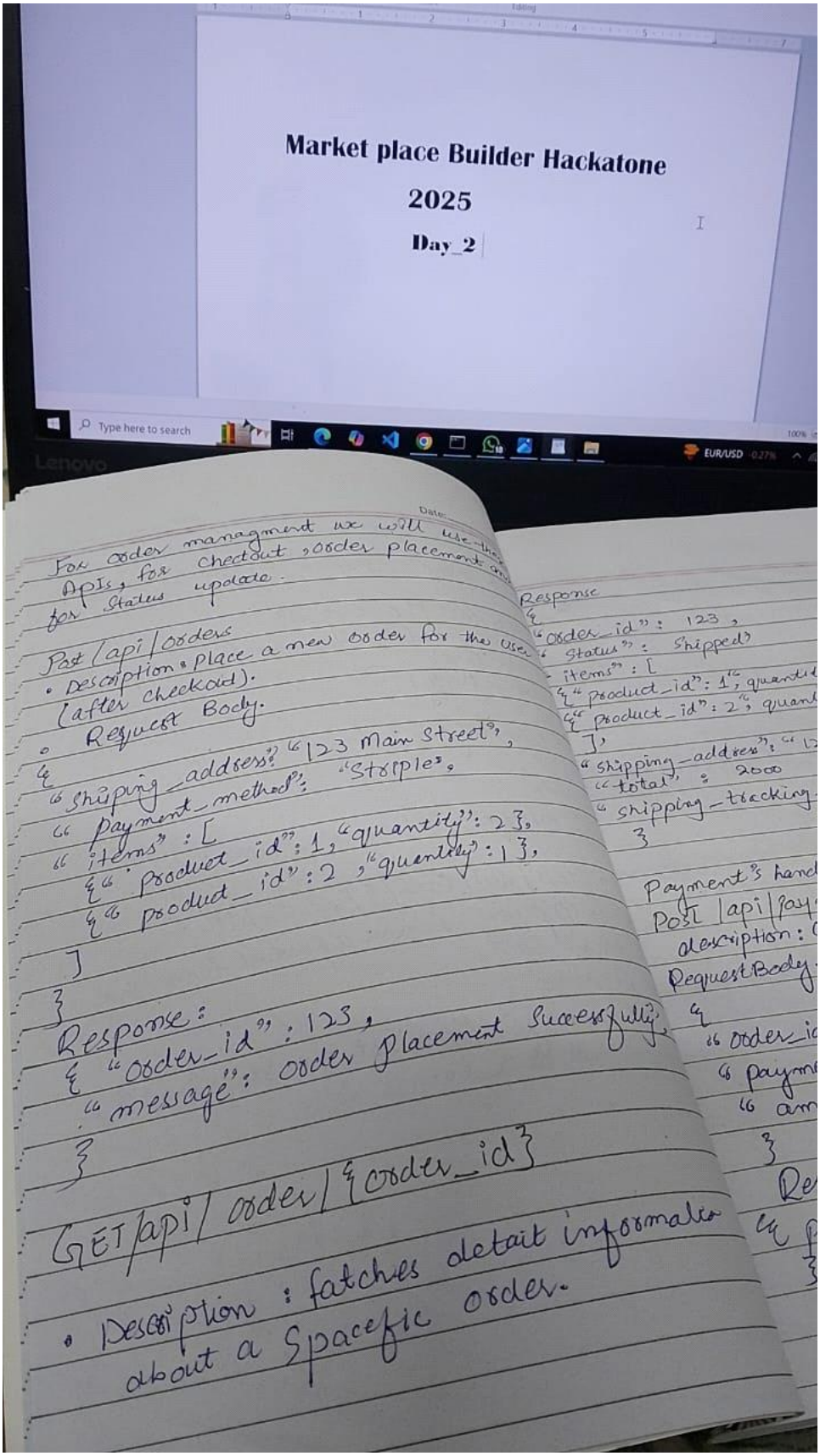
Response:

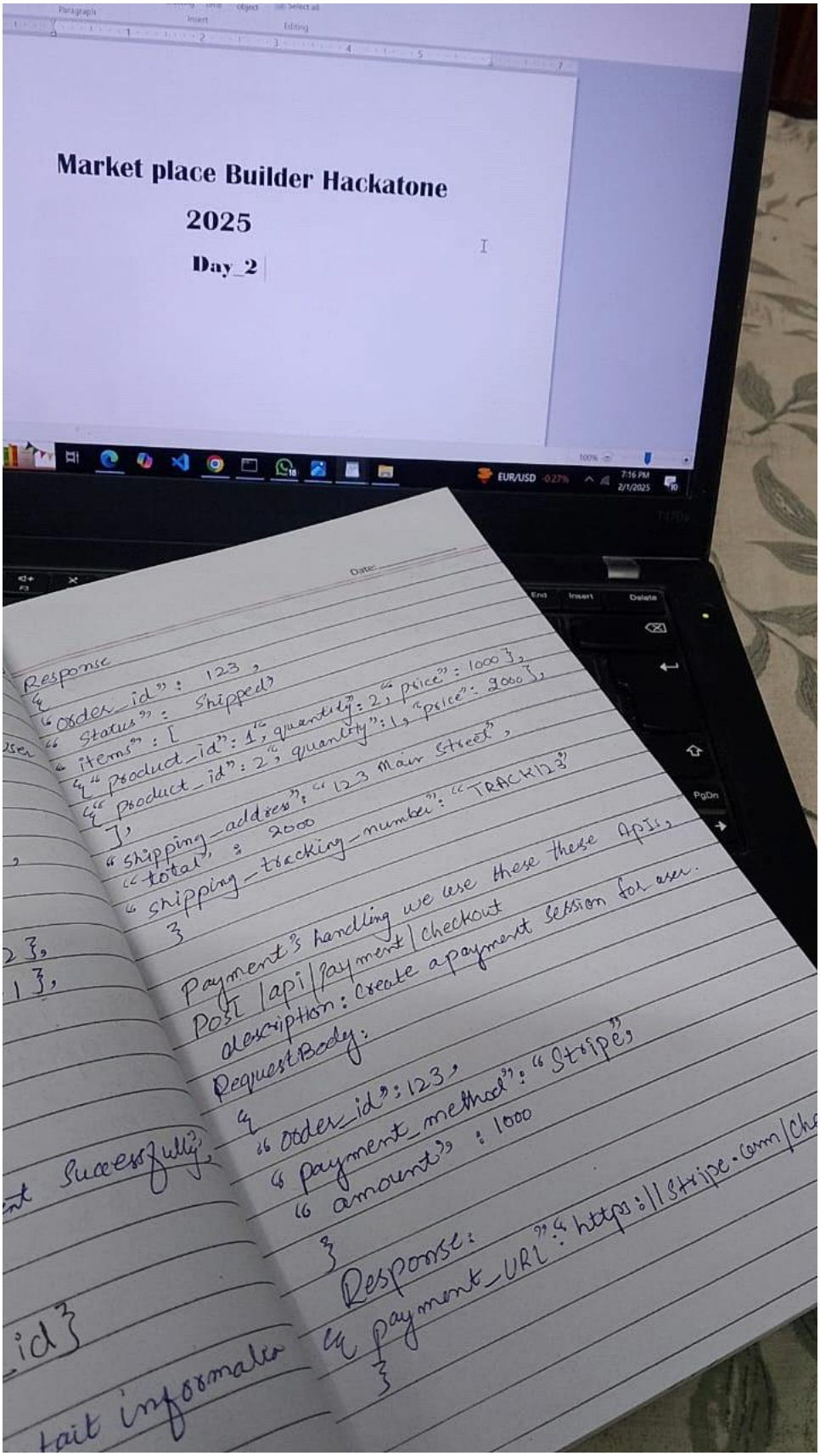

Market place Builder Hackatone
2025
Day_2
Type here to search
EUR/USD
Date:
[
{
"id" : "T-shirt",
"name" : "T-shirt"
"Price" : "1000 Rs"
"description" : "A cool cotton t-shirt",
"image-URL" : "https://express.com/t-shirt.
"Stock_quantity" : 50
"size" : ["S", "M", "L", "XL"]
}
}
]
if a admin want to add a new Produ
Post /api/Product:
•Description: new product adding
to the Categories (Admin Only).
Request Body:
"name" : "Kurti",
"Price" : "1000"
"description" : "https://express.com/kurti.jpg",
"image-URL" : "https://express.com/kurti.jpg",
"Stock_quentity" : 100,
"Sizes" : ["S", "M", "L", "XL"]
}
Response
{
"message" : produd
}
Put /api/Product/{id}
.description : update
. Request Body.
{
"Price" : 1050;
"description" : "Pri
"Stock_quantity":
}
Response:
{
"message": "Pr
}
if we dele
then
Deleted /api
• Response
{
"messa
}

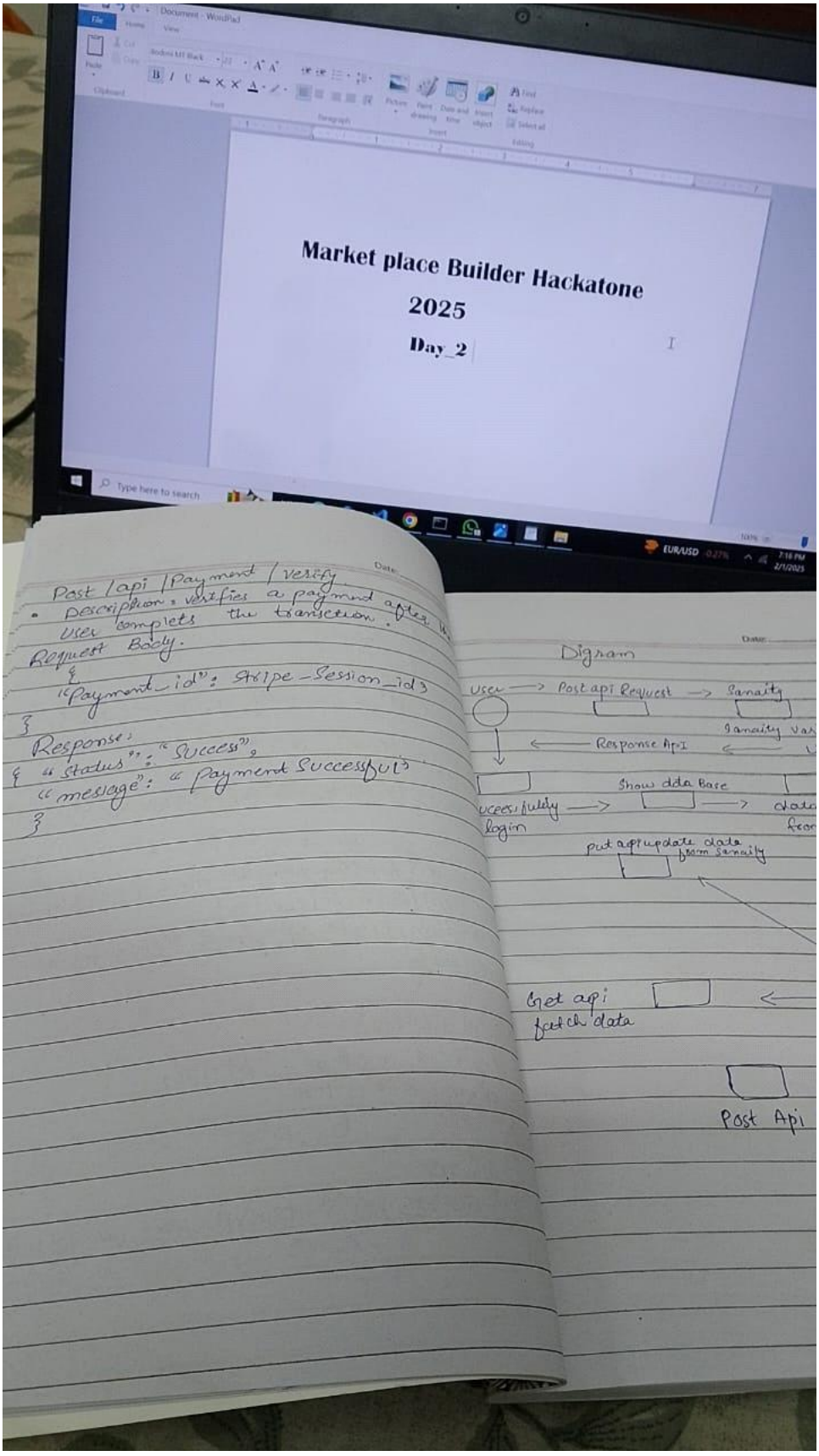

Market place Builder Hackatone
2025
Day_2
Response
{
"message": product added Successfully
}
Put /api/product/{id}
• description: update details (only admin)
• Request Body:
{
"price": 1050,
"description": "Price updated"
"stock_quantity": 80
}
Response:
{
"message": "Product updated Successfully".
}
if we deted the product (only admin)
then
Deleted /api/Product/{id}
• Response
{
"message": " Product deleted Successfully
}

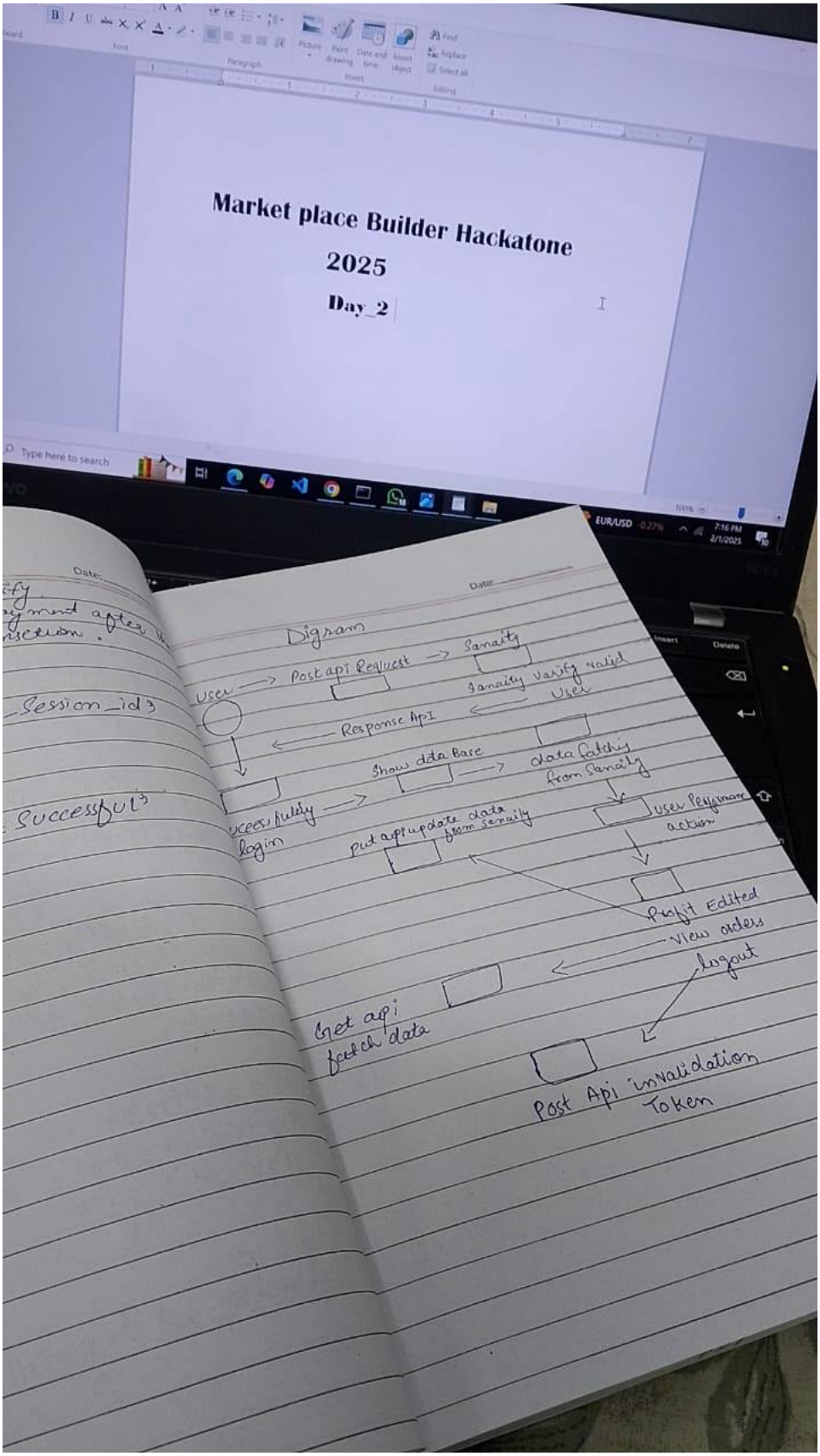

Market place Builder Hackatone
2025
Day_2
Type here to search
Lenovo
Those APIs Allow User to manage their Shopping Cart, including adding, removing and updating Items.
GET /api/Cart
• Description: place the new order for the user-card
• Request Body:
{
"item": [
{
"Product_id": 1,
"name": "T-shirt",
"quantity": 2,
"price": 1000
}
"total": 1000
}
Post /api/Cart
• Description: adds a product to the User's Cart.
• Request Body
{"Product_id": 1,
"quantity": 2
Response
{
"message": "Product
}
PUT /Api/Cart/{Product
• description: update t
in the Cart.
• Response Body:
{
"quantity":
}
Response
{"message":
}
DELETED /ap
• Descriptu
User's
• Respons
{"me
}


Market place Builder Hackatone
2025
Day_2
Response
{
"message": "Product added to Cart"
}
PUT | api | cart | {product_id}
- description: update the quantity of a product in the Cart.
- Response Body:
{
"quantity": 3
}
Response
{"message": Cart update Successfully
}
DELETED | api | cart | {product_id}
- Description: Remove a product from the user's Cart.
- Response
{"message": "Product removed from Cart"
}
to the


Market place Builder Hackatone
2025
Day_2
Type here to search
EUR/USD
For order managment we will use the
APIs, for checkout, order placement and
for status update.
Post /api/orders
• Description: Place a new order for the user
(after checkout).
• Request Body.
{
"shipping_address": "123 Main Street",
"payment_method": "Stripe",
"items": [
{"product_id": 1, "quantity": 2},
{"product_id": 2, "quantity": 1},
]
}
Response:
{ "order_id": 123,
"message": order Placement Successfully,
}
GET/api/order/{order_id}
• Description: fatches detail information
about a Spacefic order.
Response
{
"order_id": 123,
"status": "shipped"
"items": [
{"product_id": 1,
{"product_id": 2,
],
"shipping_address":
"total": 2000
"shipping_tracking
}
Payment's hand
Post /api/pay
description:
RequestBody
{
"order_id
"paym
"am
}
Re
{


Market place Builder Hackatone
2025
Day_2
Response
"order_id": 123,
"status": "shipped"
"items": [
{"product_id": 1, "quantity": 2, "price": 1000},
{"product_id": 2, "quantity": 1, "price": 2000},
],
"shipping_address": "123 Main Street",
"total": 2000
"shipping_tracking_number": "TRACK123"
}
Payment's handling we use these these apis,
Post /api/payment/checkout
description: create a payment session for user.
RequestBody:
{
"order_id": 123,
"payment_method": "Stripe",
"amount": 1000
}
Response:
{"payment_URL": "https://stripe.com/ch
}


Market place Builder Hackatone
2025
Day_2
Post /api /Payment / Verify
Description : Verifies a payment after the user completes the transaction.
Request Body.
{
"Payment_id": Stripe_Session_id
}
Response:
{ "status": "Success",
"message": "Payment Successful"
}
Digram
User
Post api Request
Sanaity
Response Api
Show data Base
Login
put appupdate data from Sanaity
Get api
fetch data
Post Api


Market place Builder Hackatone
2025
Day_2
Digram
User
Post api Request
Sanity
Sanity Verify Valid User
Response Api
Show data Base
data fetching from Sanity
Put appupdate data from Sanity
User Performs action
Profile Edited
View orders
Logout
Get api fetch data
Post Api invalidation Token
Session_id
Successful