



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment No: 2

Student Name: Keshav Datta  
Branch: BE CSE  
Semester: 6<sup>th</sup>  
Subject Name: System Design

UID: 23BCS10423  
Section/Group: 23BCS\_KRG-2\_B  
Date of Performance: 14/01/2026  
Subject Code: 23CSH-314

**1- Aim** - Design an Online shopping platform similar to Amazon / Flipkart that will allow users to purchase mobiles, laptops, cameras, clothes etc.

### 2- Requirements:

#### A- Functional Requirement

- Users should be able to search products using product name or title.
- Users should be able to view product details such as description, image, available quantity, and reviews.
- Users should be able to select quantity and add products to the cart.
- Users should be able to check out and make payments securely.
- Users should be able to check order status after placing an order.
- The system should manage limited stock inventory and handle race conditions during flash sales.

#### B- Non-Functional Requirement

- The system should support up to 100 million daily active users and handle around 10 orders per second.
- High availability is required for product search and browsing.
- Strong consistency is required for payment, order placement, and inventory management.
- The expected response time should be around 200 ms.
- The system should support horizontal and vertical scaling.

### 3- Core-entities of System

- User/Client
- Products
- Cart
- Orders
- Checkout followed by Payment

### 4- API endpoint creation

- a) GET API Call: Prod\_Search



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Https://Local\_Host/products/search\_item = {Search\_keywords}

```
HTTP Req
{
    GET: <iPhone 16>
```

```
}
```

```
HTTP Res
{
    List<ProductID:iPhone>
}
```

Now, on front-end if multiple data of respective product is coming in that case the FE becomes faulty thus ultimately increasing the Latency.

So we will be using Pagination (1,2,3,...next)

**b) GET API Call: View Product Details**

Https://Local\_Host/products/{product\_id}

```
HTTP Req
{
    GET: <Product_id=17>
}
HTTP Res
{
    Product_id=17,
    Name: iPhone17,
    Color: Navy Blue,
    Price: $1009,
    Image_URL: URL_image
}
```

**c) POST API Call: Item add in cart**

Https://Local\_Host/cart/add\_products

```
HTTP Req
{
    Product_id:17,
    Product_id:16
}
HTTP Req Header
{
    User_id: 04
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
}
```

HTTP Res

```
{
```

Cart\_id: 101

```
}
```

- d) PUT API Call: To update any order in the cart
- e) DELETE API Call: To remove any item from the cart
- f) **POST API Call: for check out & Payment**  
`Https://Local_Host/checkout -> {post body}`

```
HTTP Req
```

```
{
```

All Product Id's,  
Total Quantity,  
Total Price

```
}
```

HTTP Res

```
{
```

Order\_id

```
}
```

`Https://Local_Host/payment -> {post body}`

```
HTTP Req
```

```
{
```

Order\_id,  
Payment Type,  
Payment\_Mode

```
}
```

HTTP Res

```
{
```

Confirmation\_Status: Success / Fail

```
}
```

- g) **GET API Call: Order Status**  
`Https://Local_Host/order_status = {order_id}`

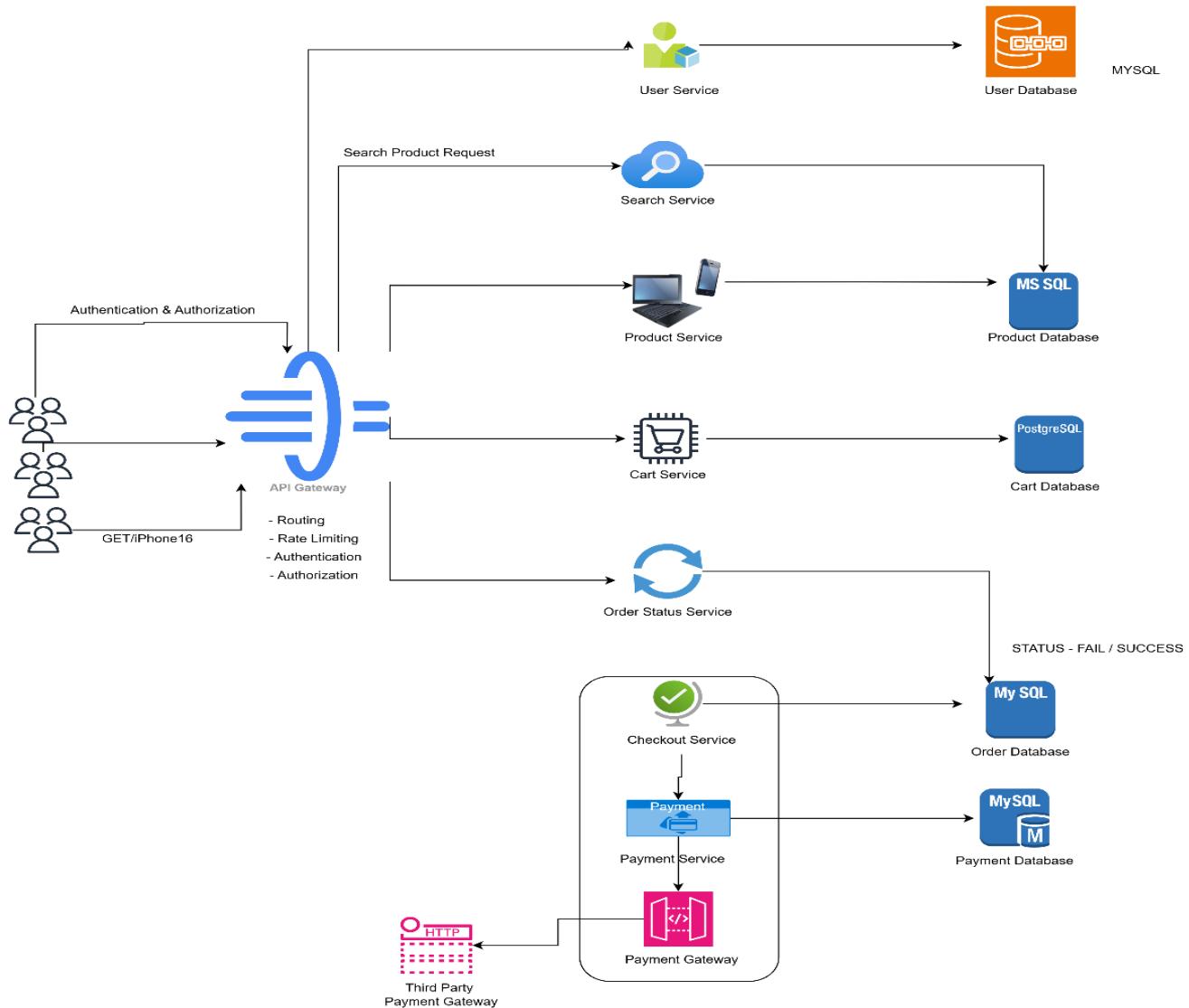
## 5- High-Level Design

Now according to the functional requirement of the system, we can identify that we have to follow a distributed / micro-services approach not the monolithic one.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



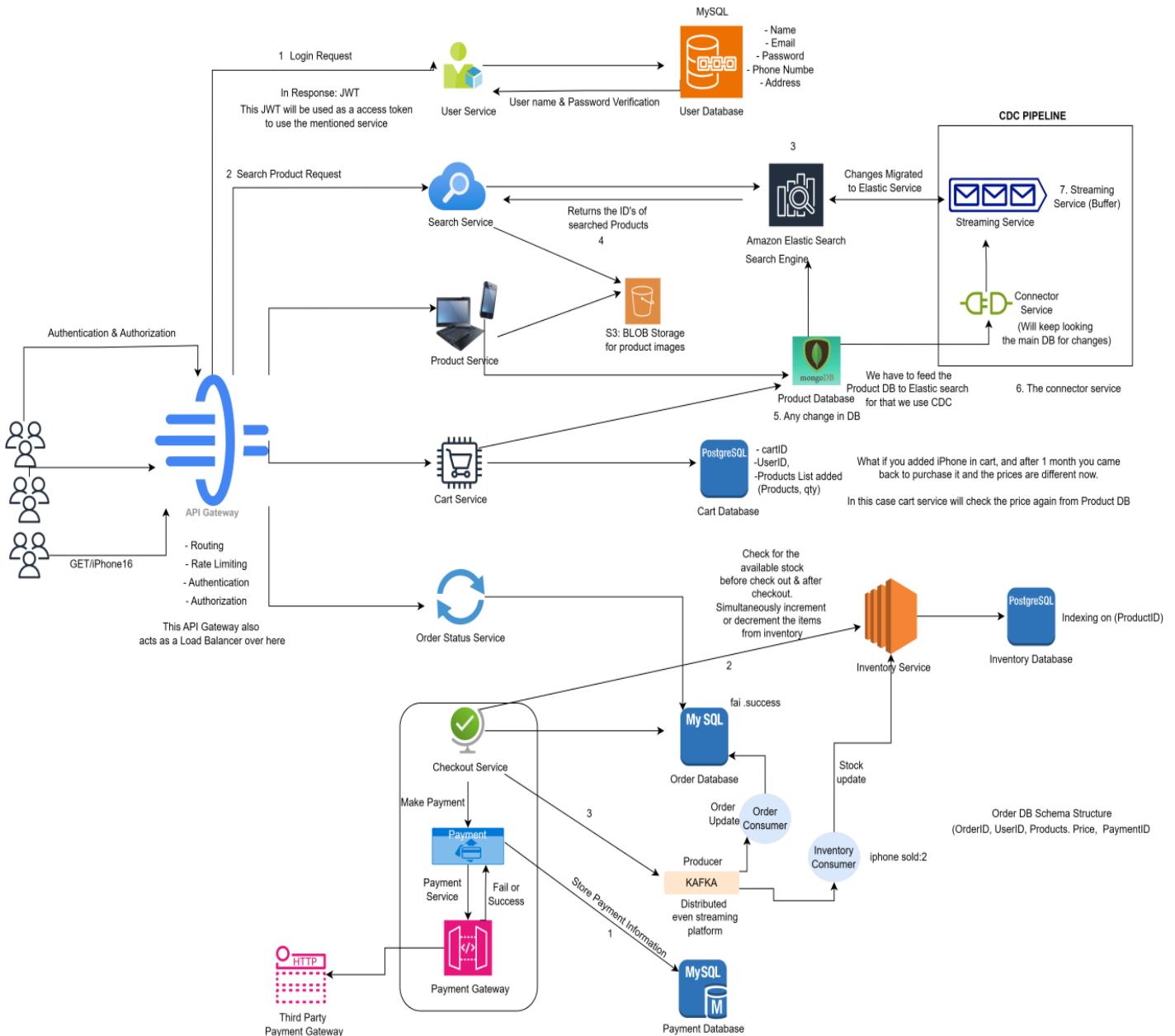
This will fulfill all the functional requirements that were listed. Now, we will see the internal implementations of each one of these components in LLD.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## 6- Low- Level Design





# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## 7. Outcome / Result

- A functional Online Shopping Platform design was successfully created.
- The system supports product search, product viewing, cart management, checkout, payment, and order tracking.
- API endpoints were designed to ensure smooth frontend–backend communication.
- Pagination was implemented to reduce latency and improve performance during product search.
- The design handles limited inventory and race conditions, ensuring correct order placement.
- The system is scalable, consistent, and reliable for high user traffic.