API / Backend Requirements Specification

1. Architecture & Components Overview

- Authentication Service: handles registration, login, password reset, token issuance.
- User Profile Service: profile updates, address management.
- Product Service: category, product listing, product details.
- Cart Service: cart CRUD operations.
- Order Service: order creation, history, status tracking.
- Payment Service: integrate external payment gateway; confirm payment.
- Notification Service: emails / SMS for OTP, order confirmations.
- Databases: RDBMS (e.g. PostgreSQL or MySQL) for core data; Redis for caching; optionally search engine (Elasticsearch) for search; queue system (RabbitMQ / Kafka) for asynchronous tasks.

2. Data Models / Schema

Table	Fields	Notes
Users	user_id, username, email, mobile, password_hash, is_email_verified, is_mobile_verified, account_locked_until, created_at, updated_at	Stores user credentials and account status
Products	product_id, category_id, name, brand, description, price, discount_price, stock_quantity, image_urls, rating_avg, rating_count, created_at, updated_at	Stores product catalog details

3. API Endpoints & Sample Request / Response

3.1 Register

Endpoint: /api/v1/auth/register

Method: POST

Authentication: None

Request:

```
{
  "username": "john_doe",
  "email": "john.doe@example.com",
  "mobile": "9876543210",
  "password": "John@123$",
  "confirm_password": "John@123$"
}
```

Success Response:

```
{
  "status": "success",
  "user_id": "uuid-1234-abcd",
  "message": "Registration successful. Please verify your email/mobile."
}
```

Error Response:

```
{
  "status": "error",
  "errors": [
    {"field": "username", "message": "Username must be 6-20 characters."},
    {"field": "password", "message": "Password must contain at least one
uppercase letter."}
  ]
}
```

3.2 Login

Endpoint: /api/v1/auth/login

Method: POST

Authentication: None

Request:

```
{
  "username_or_email": "john.doe@example.com",
  "password": "John@123$"
}
```

Success Response:

```
{
  "status": "success",
  "user_id": "uuid-1234-abcd",
  "token": "JWT_TOKEN_HERE",
```

```
  "expires_in": 3600
}
```

Error Response:

```
{
  "status": "error",
  "message": "Invalid username/email or password."
}
```

3.3 List Categories

Endpoint: /api/v1/categories

Method: GET

Authentication: Optional

Request:

```
N/A
```

Success Response:

```
{
  "status": "success",
  "categories": [
    { "category_id": "C100", "name": "Clothing" },
    { "category_id": "C200", "name": "Grocery" }
  ]
}
```

Error Response:

```
{
  "status": "error",
  "message": "Internal server error."
}
```

4. Error Codes & Response Conventions

- 200 OK: Request succeeded
- 201 Created: Resource created
- 400 Bad Request: Input validation error
- 401 Unauthorized: Missing or invalid authentication
- 403 Forbidden: Access not allowed
- 404 Not Found: Resource not found
- 423 Locked: Account locked due to failed logins
- 500 Internal Server Error: Unexpected server error