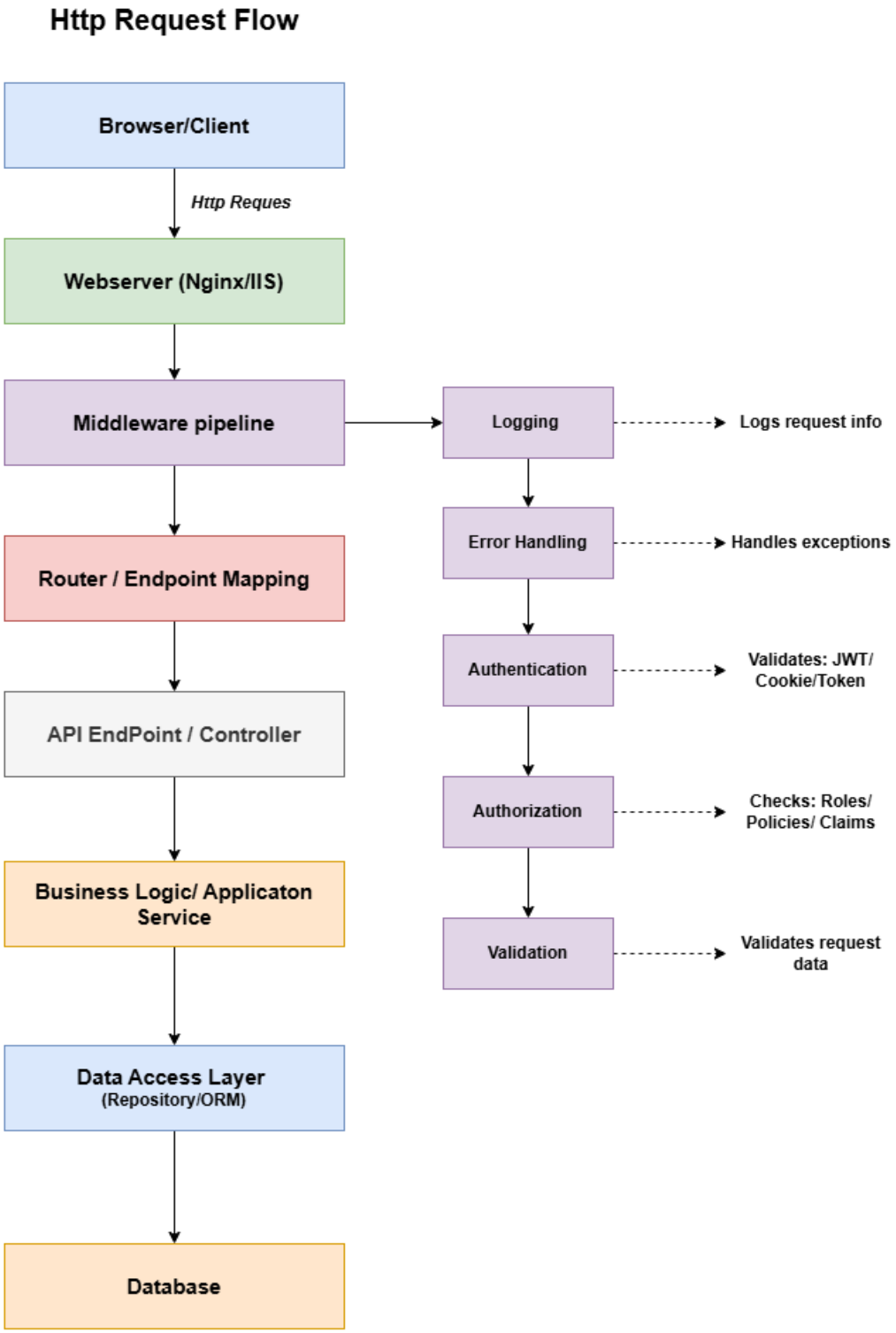
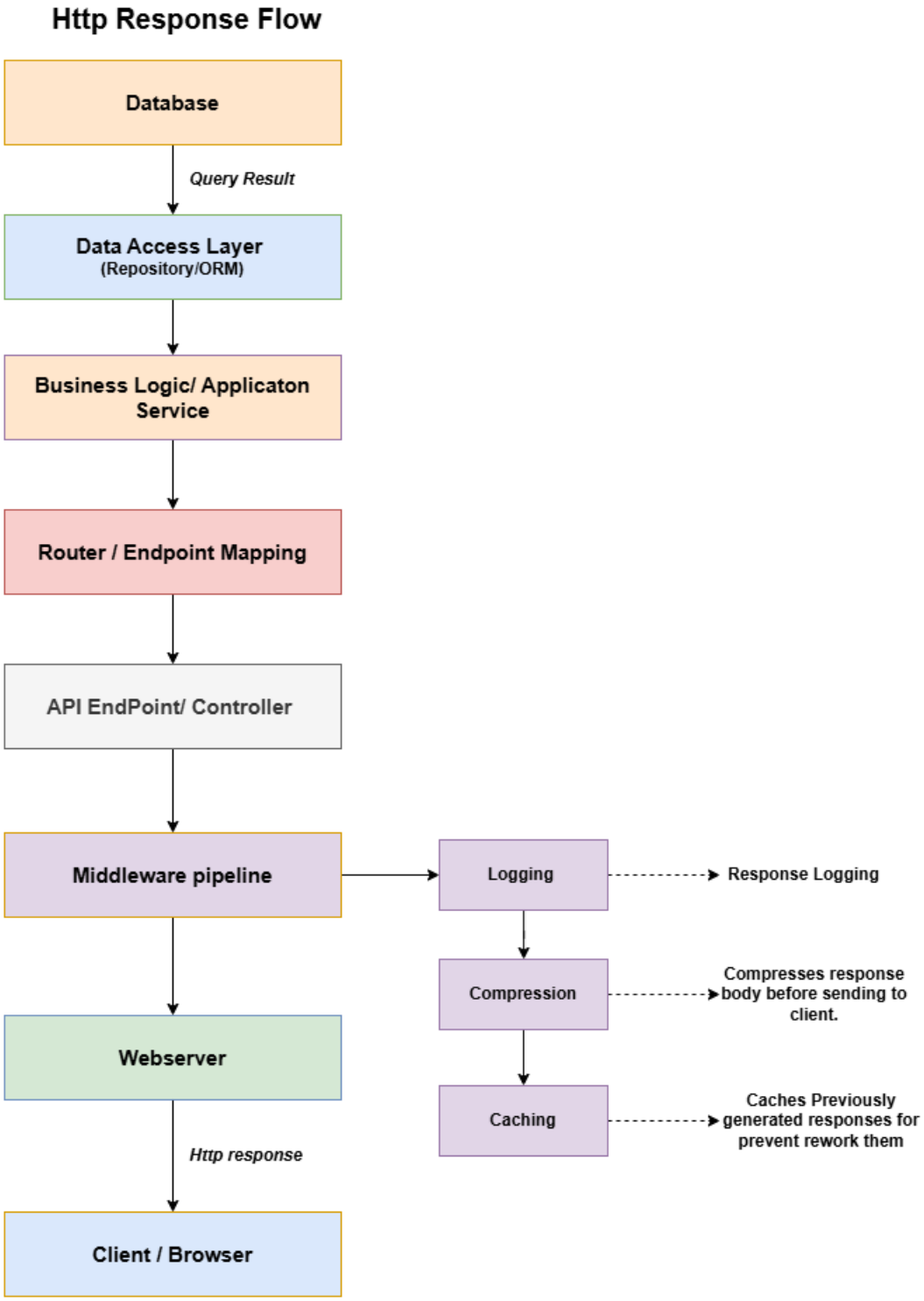
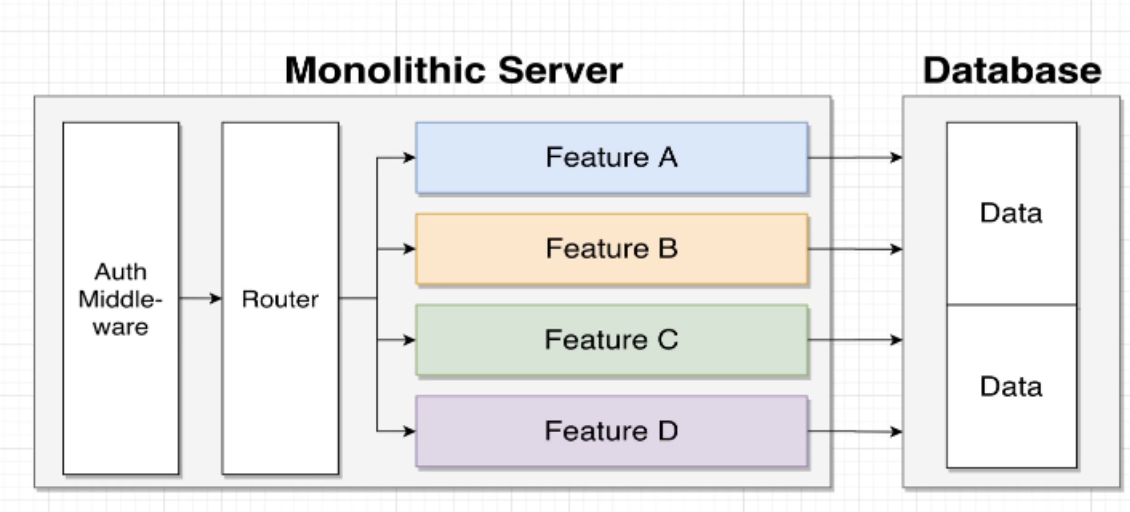
Web application Request-response flow




Http Response Flow
Database
Query Result
Data Access Layer
(Repository/ORM)
Business Logic/ Applicaton Service
Router / Endpoint Mapping
API EndPoint/ Controller
Middleware pipeline
Logging
Response Logging
Compression
Compresses response body before sending to client.
Caching
Caches Previously generated responses for prevent rework them
Webserver
Http response
Client / Browser

Monolithic Architecture



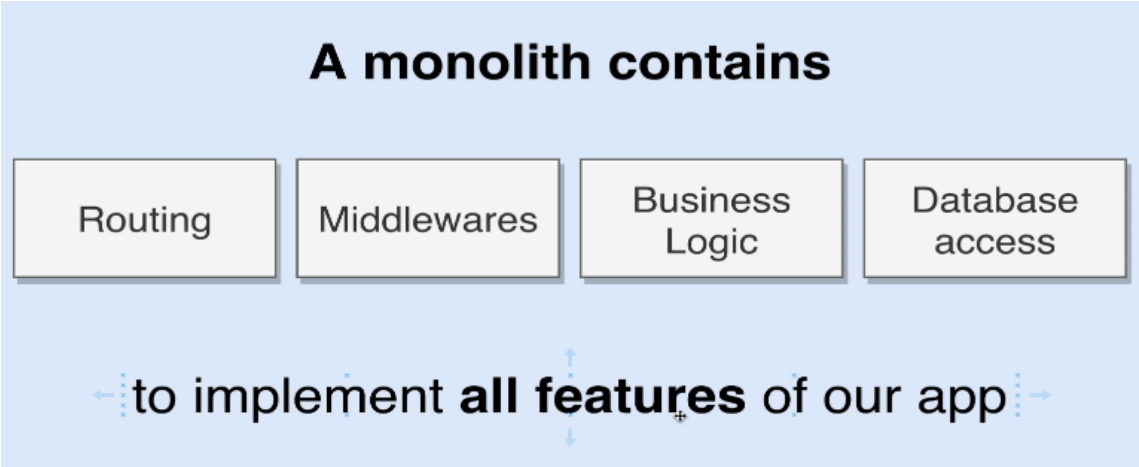
- All codes of the application implements inside a single codebase and deploy entire codebase as one discrete unit.

Let's see following scenario-

- Client send a request from browser
- That request flow through our application as follows-
    - Preprocessing middleware e.g., authentication checking, validation checking, logging etc.
    - Then goes to router
    - The router decides the feature based on the request
    - Feature read/write to database
    - After successful read/write database send a response
    - That response comes to user browser

- In Monolithic architecture –
    - All routings
    - All middleware
    - All business logics
    - All database access

  Those are require to implement for all features of our application in a single codebase

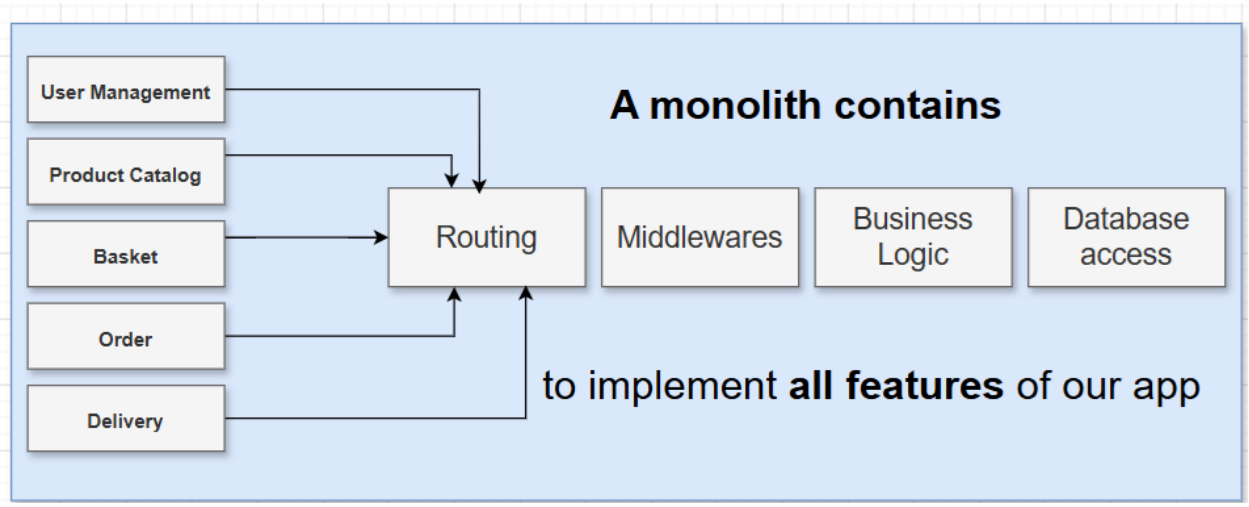


Example

We have a e-commerce web application that contains

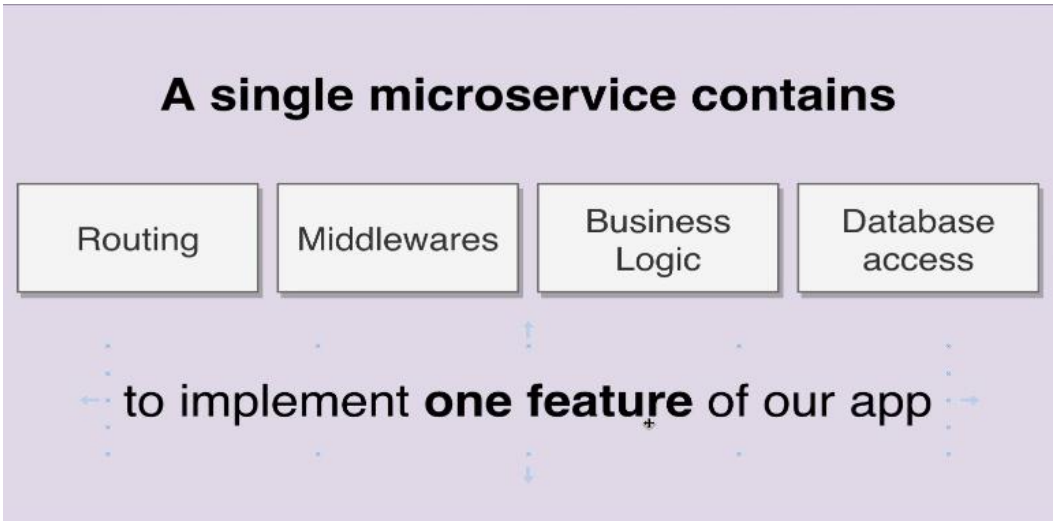
- User Management
- Product Catalog
- Basket
- Order
- Delivery

All the middleware, routing. Business logic, database accesses will be in a single codebase.

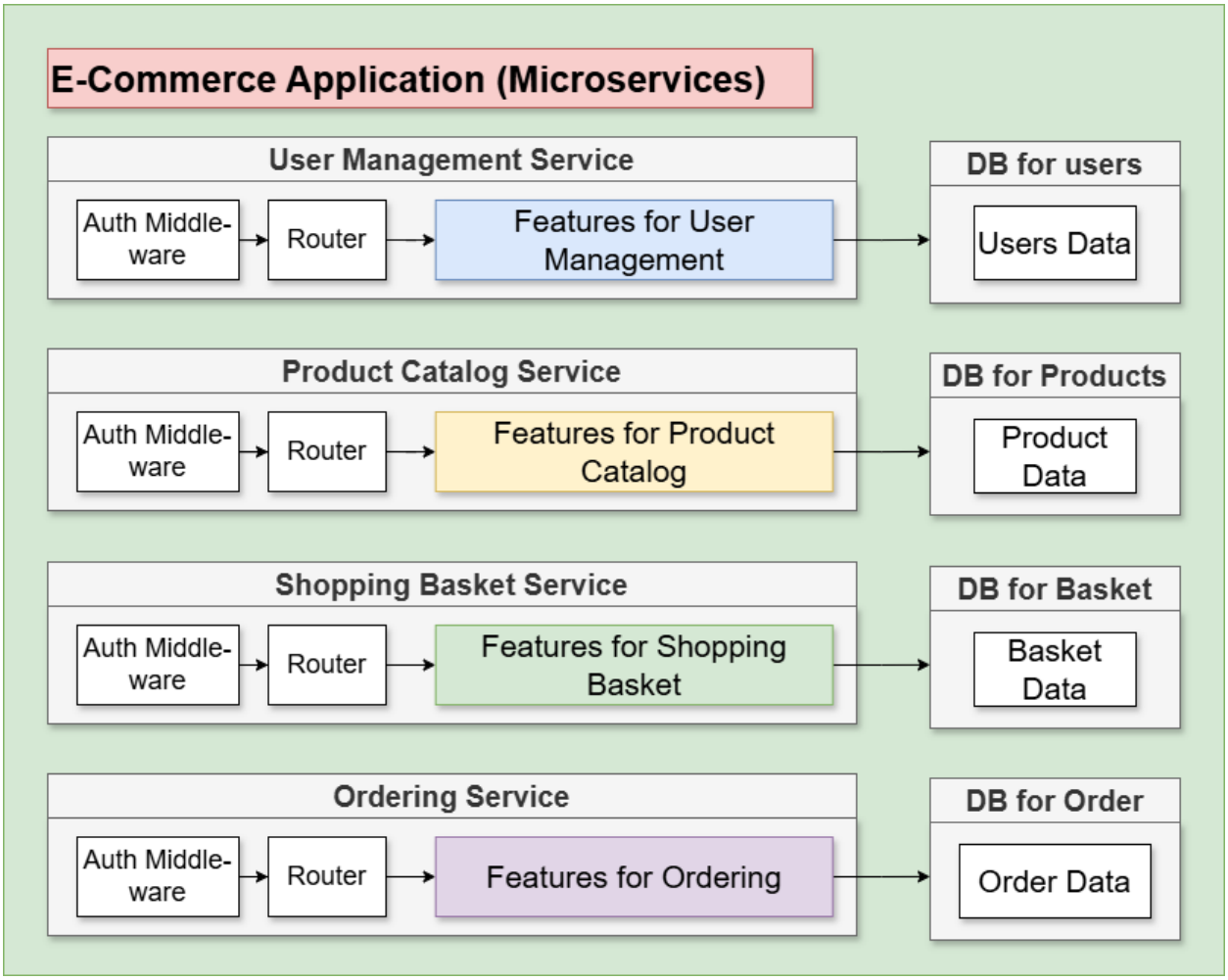


Microservices

- Microservices breaking down the large application into different services
- Each services are implement as a single unit.



- Combination of all services becomes our application.



As per above diagram-

- We split-up the entire application into different smaller and well manageable personalized services.
- Each of these services are entirely self-contained i.e., each of these services has its own-
    - Middleware
    - Router
    - Features
    - Database

  And all these together make the service work correctly.
- If for any reason any services fails therefore rest other portion of our application will work fine.