❖ Find out Domain Backend process.

1. Domain Name System (DNS) Resolution

- **Step 1.1**: **User Input**: The user enters the domain name (e.g., example.com) into a browser.
- **Step 1.2**: **DNS Lookup**: The browser sends a query to a DNS server to resolve the domain name into an IP address.
- **Step 1.3**: **Response**: The DNS server returns the IP address of the hosting server.

---

2. Server Connection

- **Step 2.1**: **Browser Requests Connection**: Using the resolved IP address, the browser sends a request to the server (via HTTP/HTTPS protocols).
- **Step 2.2**: **SSL/TLS Handshake**: If HTTPS is used, an SSL/TLS handshake ensures encrypted communication.
- **Step 2.3**: **Server Accepts Request**: The server acknowledges the request and prepares to respond.

---

3. Backend Processing

- **Step 3.1**: **Routing**:
  - The server routes the request to the appropriate backend logic based on the URL (e.g., /login, /api/data).
- **Step 3.2**: **Authentication and Authorization**:
  - The backend verifies the user's identity (via tokens, cookies, etc.) and checks permissions for the requested resource.
- **Step 3.3**: **Business Logic Execution**:
  - The backend executes the core logic (e.g., querying a database, applying algorithms, or performing calculations).
- **Step 3.4**: **Database Interaction**:
  - The backend communicates with the database to read, write, update, or delete data.
- **Step 3.5**: **Data Formatting**:
  - Retrieved or processed data is formatted into a response structure (e.g., JSON or XML).

---

4. Response to Browser

- **Step 4.1**: **Server Sends Response**:
  - The backend sends the response back to the browser. This can include HTML, CSS, JavaScript, or data for APIs.
- **Step 4.2**: **Caching**:
  - If caching is implemented, the response may be cached for faster subsequent requests.
- **Step 4.3**: **Error Handling**:
  - If an error occurs, an appropriate HTTP status code (e.g., 404, 500) and error message are returned.

---

5. Browser Renders Content

- **Step 5.1**: **Parse Response**:
  - The browser parses the HTML and requests additional resources (e.g., images, CSS, JavaScript).
- **Step 5.2**: **Execute Client-Side Logic**:
  - Any client-side scripts (e.g., JavaScript frameworks) are executed.
- **Step 5.3**: **Render Content**:
  - The browser displays the content to the user.

---

6. Continuous Monitoring and Logging

- **Step 6.1**: **Logging**:
  - Backend logs requests, errors, and performance metrics for debugging and optimization.
- **Step 6.2**: **Monitoring**:
  - Tools like New Relic or Prometheus monitor server health and performance.