Answer any three questions from the following:

Q1

a) Differentiate between Web Engineering and Web Application.

b) What do you understand by semantic and ubiquitous web applications? Give one example for each. (2+1=3)

c) Write down two functional, content, and quality requirements for a web application, such as Pipilika Search Engine. Note that the search engine has been designed based on the Bengali language to search any information.

Answer :

a) **Difference between Web Engineering and Web Application**

Aspect	**Web Engineering**	**Web Application**
Definition	A discipline for systematic, disciplined, and quantifiable approaches to developing, deploying, and maintaining high-quality web-based systems and applications.	An application software that runs on a web server and is accessed through a web browser.
Focus	Methodology, architecture, process, design, testing, and maintenance.	Providing functionalities to end users like searching, online shopping, and online banking.
Scope	Concerned with the **process and methodology** of building and managing web systems.	Concerned with the **end product** used by the user.
Users	Practiced by developers, engineers, architects, and testers.	Used by customers, clients, and general users.
Example	Engineering process used to build Google Search or Facebook.	Google Search, Facebook, Amazon, Gmail.

b) **Semantic Web and Ubiquitous Web Applications**

- **Semantic Web Application**: Applications that understand the meaning (semantics) of data and enable machines to interpret content.

    - Example: **Wolfram Alpha** (answers queries based on knowledge graph).
- **Ubiquitous Web Application**: Applications that provide seamless access anytime, anywhere, and on any device.
    - Example: **Google Maps** (works on mobile, desktop, tablets, cars, etc.).

c) **Functional, Content, and Quality Requirements for Pipilika Search Engine**

- **Functional Requirements**
    1. Must allow users to input queries in Bengali.
    2. Must retrieve relevant search results from the index.
- **Content Requirements**
    1. Should support Bengali news, blogs, educational content.
    2. Should update the index with fresh content regularly.
- **Quality Requirements**
    1. Must provide results within 2–3 seconds (performance).
    2. Must ensure availability 24/7 (reliability).

Q2

a) Write down the principles of web engineering.
b) Draw a use case diagram for an online food ordering system, where three actors are involved, such as admin, user, and delivery man.
c) In web engineering, how do you think legacy systems affect the requirement engineering process?

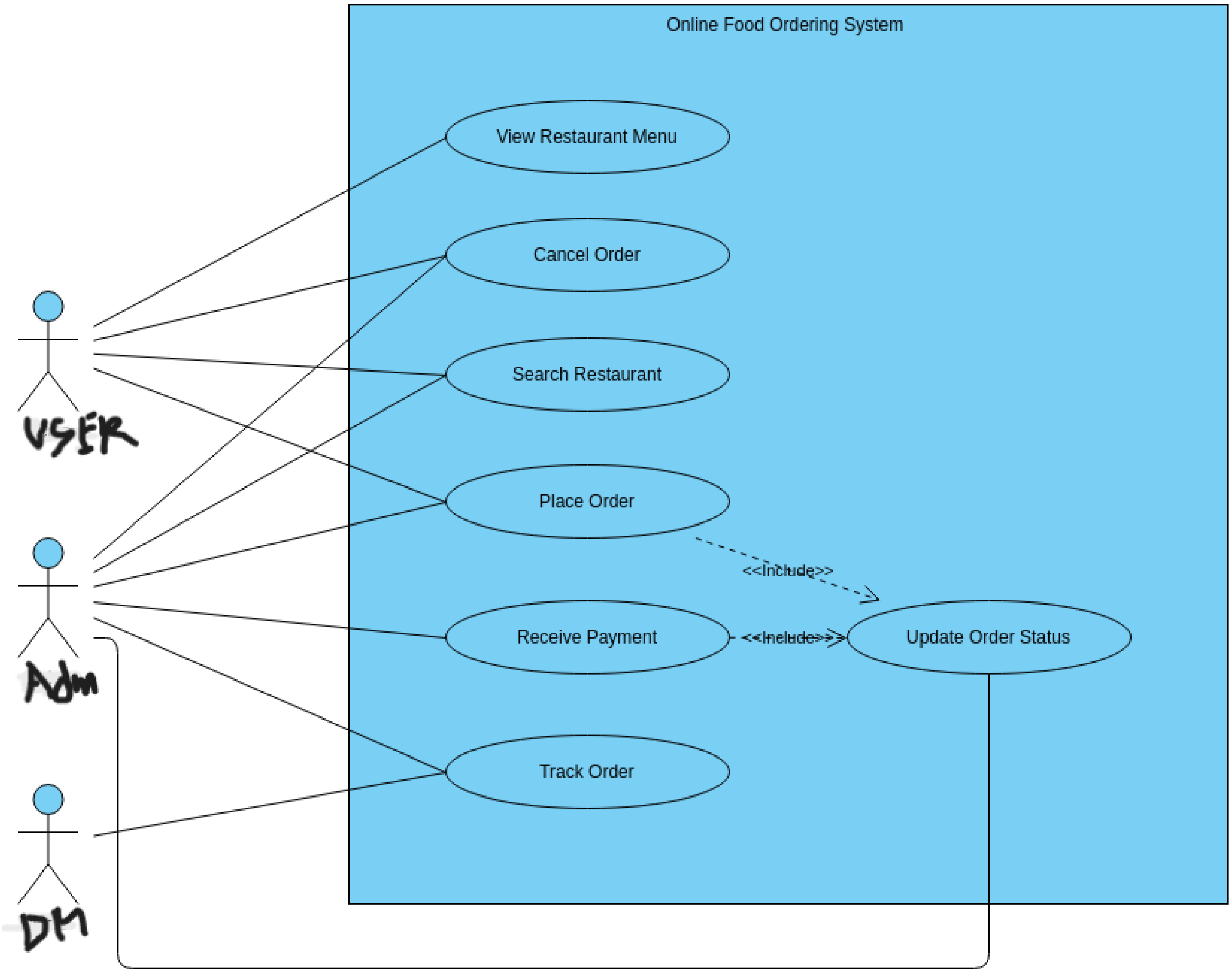
<u>Answer </u>:

a) Principles of Web Engineering

1. Customer-centric development.
2. Incremental and iterative development.
3. Continuous evolution (web apps are never “finished”).
4. Reusability of components.

5. Quality assurance (performance, usability, security).
6. Multidisciplinary approach (design, content, UI, software).

b) **Use Case Diagram - Online Food Ordering System**
Actors: **Admin, User, Delivery Man**



c) Legacy Systems and Requirement Engineering

- Legacy systems are old software or databases still in use.
- They affect requirements engineering because:
    1. New web applications must integrate with existing legacy systems.
    2. Constraints from outdated technology may limit new requirements.
    3. Data migration and compatibility issues arise.

Q3

a) Define web page and web site. How do you distinguish between software and web applications?
b) Write down two product-related and two usage-related characteristics for an online food review website.
c) Give two examples of workflow-based web applications.

Answer :

a) Definitions

- **Web Page**: A single document on the web, accessible via a unique URL (e.g., a news article).
- **Web Site**: A collection of interconnected web pages under a common domain (e.g., www.bbc.com).
- **Software vs Web Application**

Aspect	Software Application	Web Application
Run on	Local computer/device	Web server via browser
Installation	Needs installation	No installation
Platform	Platform-dependent	Platform-independent
Updates	Manual updates	Auto updates (server-side)
Internet	Works offline	Needs internet
Examples	MS Word, Photoshop	Gmail, YouTube

b) **Product- and Usage-related Characteristics for Online Food Review Website**

- **Product-related**
  1. Must allow users to post reviews and ratings.
  2. Must support images of food items uploaded by users.
- **Usage-related**
  1. Easy navigation and mobile-friendly UI.
  2. Search and filter reviews by restaurant, location, or rating.

c) Workflow-based web applications are systems designed to automate and manage a series of tasks or processes in a defined sequence. They ensure that each step in a process is completed before moving to the next, often involving multiple users with different roles.

Examples:

1. **Online Job Application System**
   - **Workflow:**
     1. Candidate submits an online application.
     2. HR reviews the application.
     3. Shortlisted candidates are scheduled for an interview.
     4. Interview feedback is submitted.
     5. Job offer is made to selected candidates.
   - **Explanation:** This system ensures that each step is completed in order, automates notifications, and tracks the progress of each application.
2. **Online Leave Management System**
   - **Workflow:**
     1. Employee applies for leave via the web portal.
     2. Manager reviews and approves or rejects the request.
     3. HR updates the leave records.
     4. Leave status is notified to the employee.
   - **Explanation:** This system streamlines leave requests, prevents conflicts, and maintains accurate records without manual paperwork.

Q4

a) What do you understand by requirements? Where do requirements come from?
b) How do you distinguish the requirements engineering between conventional

software and a web application? Describe briefly.

c) Draw an activity diagram for an online library management system.

Answer :

a) Requirements are statements that describe what a system should do (functional) or qualities it should have (non-functional). They define the needs and expectations of users for a software or web application.

Sources of Requirements:

1. **Stakeholders** – Users, clients, managers, and developers.
2. **Business Rules** – Organizational policies and regulations.
3. **Existing Systems** – Legacy systems or similar applications.
4. **Standards & Guidelines** – Industry or technical standards.
5. **Market Research** – User surveys, competitor analysis, and trends.

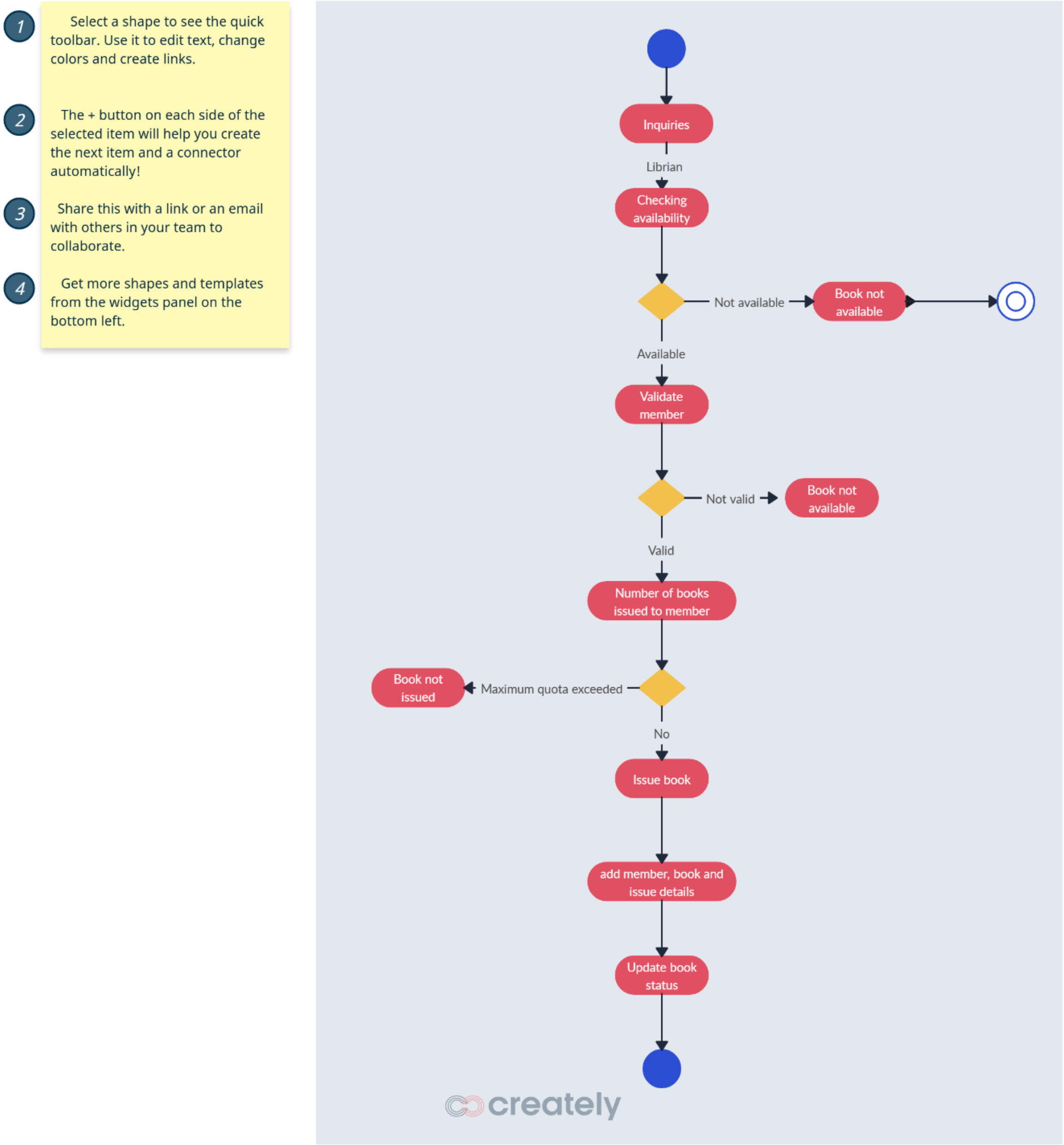
b) **Conventional Software vs Web Application:**

Aspect	**Conventional Software**	**Web Application**
Environment	Runs on a single platform or OS.	Runs on multiple platforms via web browsers.
User Interaction	Limited, often local users.	Large, diverse, and remote user base.
Performance & Scalability	Moderate; usually predictable usage.	High; must handle variable traffic and concurrent users.
Security & Privacy	Important but less exposed to internet threats.	Critical; exposed to web attacks and public access.
Update & Maintenance	Updates need manual installation.	Updates are centralized and instant for all users.

Aspect	**Conventional Software**	**Web Application**
Requirement Changes	Relatively stable once deployed.	Frequently evolving due to changing user needs and web standards.

Summary:
Web applications require more attention to scalability, cross-platform compatibility, security, and frequent updates, whereas conventional software focuses more on local functionality and stability.

c) An activity diagram for an online library management system:



1 Select a shape to see the quick toolbar. Use it to edit text, change colors and create links.

2 The + button on each side of the selected item will help you create the next item and a connector automatically!

3 Share this with a link or an email with others in your team to collaborate.

4 Get more shapes and templates from the widgets panel on the bottom left.

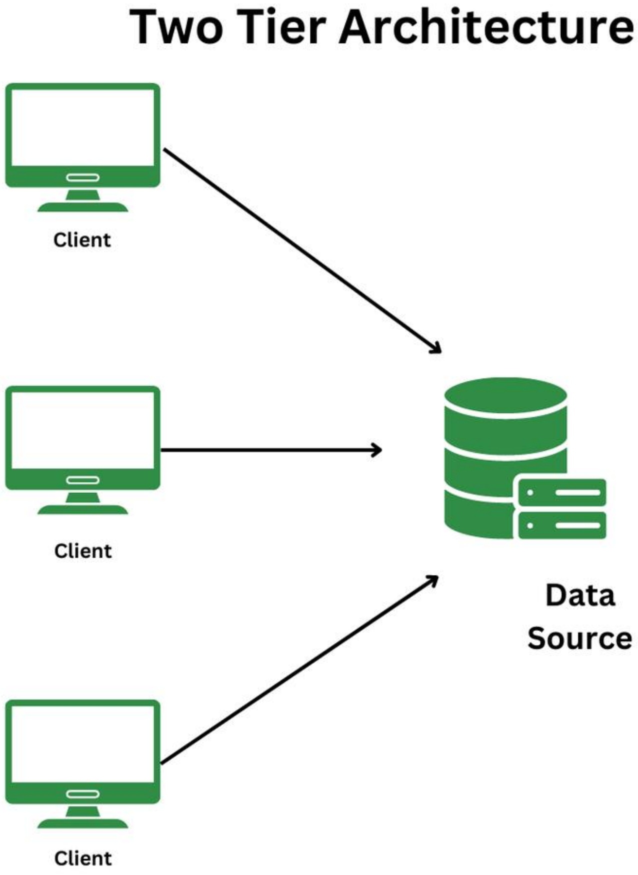
Answer all the questions from the following:

1. Answer any five:

a) Draw the 2-layered architecture for web applications.
b) What is the role of the developer and the tester in web application development?

c) Write the full forms of DOM, VSM, SMTP, RTSP.
d) What are the steps followed in load testing?
e) What is hypertext?

Answer :



a)

b) **Role in Web Application Development**

Role	**Responsibilities**
Developer	Designs and writes code, integrates databases, implements functionality, ensures the application works as intended.
Tester	Validates functionality, finds and reports bugs, checks usability, performance, and security, ensures quality before deployment.

C) **Full Forms**

Abbreviation	**Full Form**
DOM	Document Object Model
VSM	Vector Space Model
SMTP	Simple Mail Transfer Protocol
RTSP	Real-Time Streaming Protocol

d) **Steps in Load Testing:**

1. **Identify critical scenarios** – Determine which parts of the web application need performance testing.
2. **Design test scripts** – Create scripts that simulate real user interactions.
3. **Execute tests** – Run the application under expected or high load conditions.
4. **Monitor performance** – Measure CPU, memory, response time, and throughput.
5. **Analyze results** – Identify bottlenecks and optimize performance.

e) **Hypertext**

- **Definition:** Text that contains **links (hyperlinks)** to other documents, allowing **non-linear navigation** on the web.
- **Example:** Clicking a news headline on a website to open the full article.

2. Answer any four:

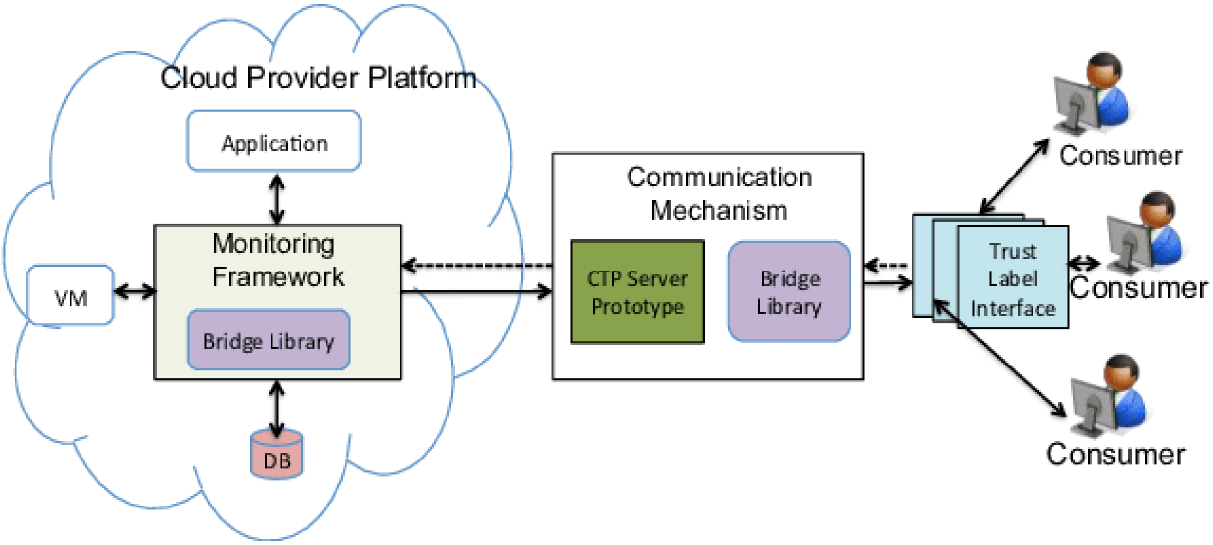
a) State the factors influencing the development of a web application architecture.
b) Draw the integration architecture for a web application.
c) Give definition of idioms, design, and architecture patterns.
d) Draw a diagram to show the phases in the WebML development process.

Answer :

a) **Factors Influencing Web Application Architecture**

1. **Scalability** – Ability to handle growing number of users or data.
2. **Performance** – Fast response times and efficient resource usage.
3. **Security** – Protection against threats like hacking, data breaches, and unauthorized access.
4. **Integration** – Compatibility with legacy systems, databases, and third-party services.
5. **User Experience** – Ease of use, responsiveness, and accessibility across devices.
6. **Technology Stack** – Choice of programming languages, frameworks, and tools.

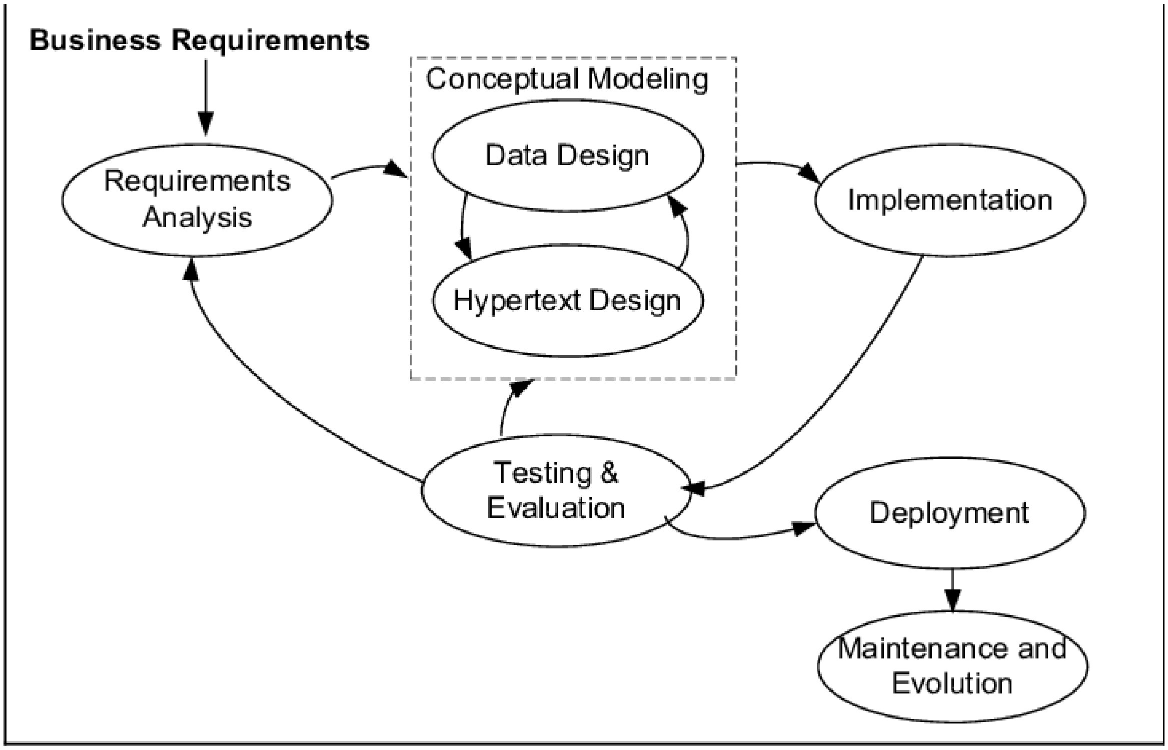
b) Integration architecture for a web application:



c) **Definitions**

Term	**Definition**	**Example**
Idioms	Low-level coding practices or conventions specific to a programming language.	Using Python's list comprehension for loops.
Design Patterns	Reusable solutions to common software design problems.	MVC (Model-View-Controller), Singleton.
Architecture Patterns	High-level structural templates guiding the organization of an entire system.	3-tier architecture, Client-Server architecture.

d)



The phases in the WebML development process:

1. **Requirement Analysis** – Identify what the web application should do.
2. **Data Design** – Model the data and relationships.
3. **Hypertext Design** – Structure the navigation and links.
4. **Presentation Design** – Design the UI and layout.
5. **Implementation & Deployment** – Develop, test, and release the web application.

---

3. Answer any two:

a) What do you understand by pattern and framework?
b) Draw the basic components of web application architecture and show their relation.
c) State and explain any four aspects of web application security.

Answer :

a) What do you understand by pattern and framework?

- A **pattern** is like a reusable *idea or template* for solving a recurring problem. Example: MVC (Model–View–Controller) is a pattern that separates logic, interface, and data.
- A **framework** is a ready-made *software structure or toolkit* that provides libraries, components, and rules to build applications. Example: Django (Python) or Spring (Java).

b) Basic components of web application architecture

A typical web app has 3 layers:

1. **Client (Browser/Front-end)** → sends requests, displays UI.
2. **Web Server (Middle layer)** → processes requests, runs the app logic.
3. **Database Server (Back-end)** → stores and retrieves data.

Relations:

- Client ↔ Web Server ↔ Database.
- Sometimes a **Load Balancer** and **API services** are in between.

c) Four aspects of web application security

1. **Authentication** – verifying the user (e.g., login with password, OTP).
2. **Authorization** – controlling what the user can access (e.g., admin vs. regular user).
3. **Data Encryption** – protecting data in transit (HTTPS, SSL/TLS) and at rest.
4. **Input Validation** – preventing attacks like SQL injection or XSS by sanitizing user input.

Other possible aspects: session management, error handling, secure APIs, etc.