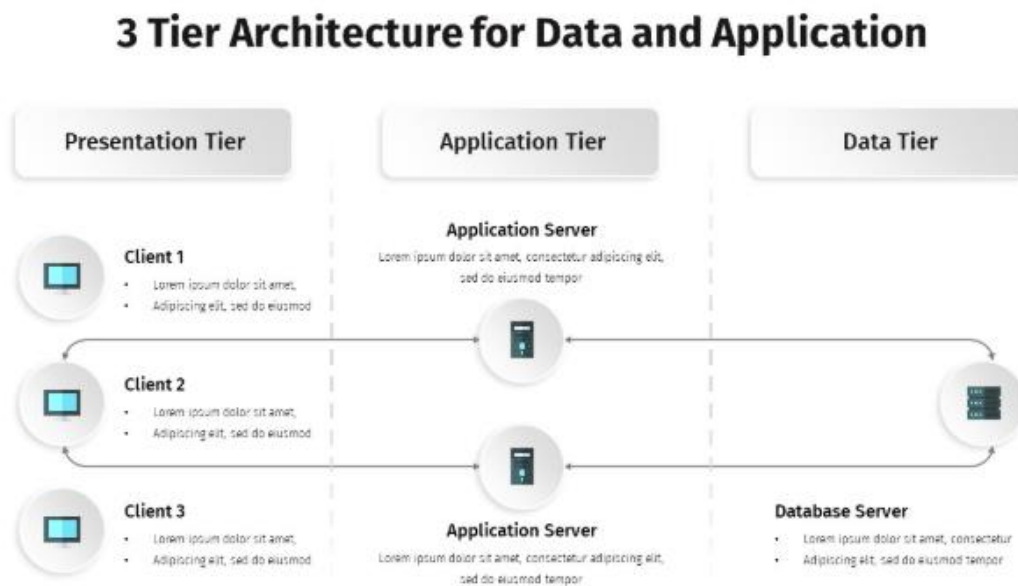


Practical No. 1 :- Three tier of architecture

The three-tier architecture is a widely used design pattern in software development, especially for web and enterprise applications. It separates an application into three distinct layers or tiers, each with specific responsibilities. This modular approach improves scalability, maintainability, and flexibility. Here's a detailed explanation of the three tiers:



1. Presentation Tier (User Interface Layer)

Purpose:

- This is the topmost layer where the application interacts with the end user. It provides the user interface (UI) that users see and interact with to input data and view results.

Responsibilities:

- Accept input from users (e.g., form data).
- Display data and results in a user-friendly format (e.g., HTML, dashboards, charts).
- Communicate with the middle tier (business logic layer) to fetch or send data.

Components:

- Web browsers or mobile apps for client-side rendering.
- Front-end technologies like:
 - HTML, CSS, JavaScript, React, Angular, or Vue.js (for web apps).
 - Swift, Kotlin, or Flutter (for mobile apps).

Examples:

- Login pages, shopping cart interfaces, and dashboards in web or mobile apps.

2. Business Logic Tier (Application Layer)

Purpose:

- This is the middle layer where the application's core functionality is implemented. It handles the logic of processing user requests, applying business rules, and making decisions.

Responsibilities:

- Process user inputs from the presentation layer.
- Apply business logic and workflows.
- Communicate with the data layer to retrieve or store information.
- Ensure security, authentication, and data validation.

Components:

- Application servers or APIs for processing logic.
- Programming languages/frameworks like:
 - Java, Python, Node.js, .NET, Ruby on Rails, or PHP.
 - Frameworks such as Spring Boot, Django, Express.js, or Laravel.

Examples:

- Calculating discounts in an e-commerce application.
- Validating user credentials during login.
- Generating dynamic reports based on user queries.

3. Data Tier (Database Layer)

Purpose:

- This is the bottom layer responsible for storing, retrieving, and managing the application's data. It serves as the foundation for all data operations.

Responsibilities:

- Store structured or unstructured data persistently.
- Respond to queries from the business logic tier.
- Ensure data security, backup, and recovery.
- Optimize data storage and retrieval for performance.

Components:

- Databases like:
 - Relational Databases (RDBMS): MySQL, PostgreSQL, SQL Server, Oracle.
 - NoSQL Databases: MongoDB, Cassandra, DynamoDB.
- Database Management Systems (DBMS) and tools for querying and managing data (e.g., SQL).

Examples:

- Storing user account details, product inventory, or transaction records.
- Handling complex queries like retrieving all orders placed by a specific user.

How the Three Tiers Work Together:

1. User Request:
 - A user interacts with the presentation tier (e.g., submits a form or clicks a button).
2. Processing:
 - The presentation tier sends the request to the business logic tier, which processes the request according to the application's business rules.
3. Data Handling:

- The business logic tier interacts with the data tier to retrieve or update the required data.
4. Response:
- The retrieved or processed data flows back from the data tier to the business logic tier, which sends it to the presentation tier to display the result to the user.

Advantages of Three-Tier Architecture:

1. Scalability:
 - Each tier can be scaled independently (e.g., adding more servers for the database or application).
2. Maintainability:
 - The modular structure makes it easier to update or modify specific layers without affecting the others.
3. Flexibility:
 - Technologies in each tier can be replaced or upgraded independently.
4. Security:
 - Sensitive data and business logic are kept on separate tiers, reducing exposure to unauthorized users.

Real-World Examples:

1. E-commerce Website:
 - Presentation Tier: User browses products and adds them to the cart.
 - Business Logic Tier: Calculates the total price, applies discounts, and processes payment.
 - Data Tier: Stores product details, inventory, and user transaction data.
2. Banking Application:
 - Presentation Tier: User views account balance via a mobile app.

- Business Logic Tier: Authenticates the user and retrieves account data.
- Data Tier: Stores account information, transaction history, and customer details.