These questions were asked in previous sessions

Instruction: Don't limit your preparation to just these questions, prepare according to your project.

1. Introduction of the Project

- A **project introduction** gives a brief overview of your system, its purpose, scope, and objectives.
  Example: *This project is a Daycare Management System designed to help administrators and staff efficiently manage daily child activities, attendance, reports, and media uploads.*

2. What is Waterfall Model and Spiral Model?

- **Waterfall Model**: A linear sequential model where each phase depends on the deliverables of the previous one.
- **Spiral Model**: A risk-driven model that combines iterative development (prototyping) and systematic aspects (waterfall).

3. What is Methodology of these models in Project?

- **Waterfall Methodology**: Phases like Requirement, Design, Implementation, Testing, and Maintenance flow in order.
- **Spiral Methodology**: Repeats in loops (spirals), each consisting of planning, risk analysis, engineering, and evaluation.

4. Waterfall Model Functional and Non-Functional Requirements

- **Functional Requirements**: What the system should do (e.g., login, upload activity).
- **Non-Functional Requirements**: How the system performs (e.g., security, usability, performance).

5. VU Process Model

VU (Virtual University) might follow a tailored software process model for academic purposes, often similar to a modified waterfall or iterative model that supports gradual development.

6. What is Use Case Diagram?

A **Use Case Diagram** represents the functionality of a system from a user's perspective using **actors** and **use cases**.

7. What is Extend and Include in Use Case Diagram?

- **Include**: Extracts common behavior into a shared use case (mandatory).
- **Extend**: Represents optional or conditional behavior that may extend the base use case.

8. What is Actor, Admin, and Client and Their Functionality?

- **Actor**: Interacts with the system (e.g., Admin, Client).
- **Admin**: Manages users, data, settings.
- **Client**: Views reports, receives updates, submits information.

9. What is Architectural Design?

It defines the overall structure of the system including components, their interactions, and technologies used.

10. Types of Tiers and Their Functionality

- **1-Tier**: All layers in one machine.
- **2-Tier**: Client-Server architecture.
- **3-Tier**: Presentation (UI), Business Logic, Data Layer.

11. What is Sequence Diagram?

Shows object interactions in time sequence, emphasizing message flow between objects.

12. Lifeline in Sequence Diagram

A vertical dashed line representing the existence of an object over time.

- **Starts**: When an object is created.
- **Ends**: When the object is destroyed or the interaction ends.

13. What is Class Diagram?

Shows the static structure of a system: classes, attributes, operations, and relationships.

14. What is ER Diagram?

An **Entity-Relationship Diagram** models the data structure using entities, attributes, and relationships.

15. What is Entity?

An object or thing in the real world with a distinct existence (e.g., Child, Staff).

16. What are Attributes?

Properties or characteristics of an entity (e.g., Child_Name, Age).

17. Function of Diamond Symbol?

Used in ER diagrams:

- **Empty Diamond**: Aggregation.
- **Filled Diamond**: Composition.

18. Function of Oval Symbol

In ER diagrams: Represents **attributes**.

19. Function of Rectangular Symbol

In ER diagrams: Represents **entities**.

20. What is Database Diagram?

Visual representation of a database schema: tables, relationships, keys, and constraints.

21. What is Relationship?

Defines how entities are related (e.g., one-to-many, many-to-many).

22. Database Diagram: Use of Primary and Foreign Key

- **Primary Key**: Uniquely identifies each record.
- **Foreign Key**: Refers to a primary key in another table to establish a link.

23. What is Aggregation?

A type of relationship representing a whole-part (has-a) relation; the part can exist independently.

24. Types of Messages in Sequence Diagram

- **Synchronous**: Caller waits for response.
- **Asynchronous**: Caller proceeds without waiting.

25. What is Association?

A relationship between two classes in UML indicating how they interact.

26. What is Inheritance?

A mechanism where one class inherits attributes and methods from another class.

27. What is Cardinality?

Specifies the number of instances in one entity that can be associated with instances in another entity (1:1, 1:N, M:N).

28. What are Entities and Attributes?

- **Entities**: Real-world objects.
- **Attributes**: Data associated with entities.

29. What is Sequence Diagram?

Visualizes object interactions arranged in time sequence.

30. Synchronous or Asynchronous Message

- **Synchronous**: Blocking call with a return (e.g., function call).
- **Asynchronous**: Non-blocking call (e.g., event notification).

31. Use Case Extend and Include

- **Include**: Reuse of behavior (compulsory).
- **Extend**: Optional extension of base behavior.

32. What is System Boundary in Use Case?

A box representing the scope of the system. Use cases are inside; actors are outside.

33. How to Interact Object in Sequence Diagram

Objects interact via **messages** (arrows) that show method calls and responses.

34. What are Actors?

Users or systems that interact with your system.

35. What is System Boundary in Use Case Diagram?

Graphical box that defines what is inside or outside the scope of the system.

36. What is Lifeline?

The existence of an object over time in a sequence diagram.

37. What is the Purpose of Your Project?

To simplify daycare activity management, ensuring better communication, documentation, and efficiency.

38. What is Use Case Diagram and Why We Use It?

Used to capture functional requirements and show interactions between actors and use cases.

39. Include and Extend: What is the Difference?

- **Include**: Shared use case reused in others.
- **Extend**: Optional behavior triggered under certain conditions.

40. Oval Shape Represents Which Thing?

In use case diagrams: **Use cases**.

41. How Do We Represent Return Message?

Dashed arrow pointing back to the caller in a sequence diagram.

42. What is Activation Box and Return Message?

- **Activation Box**: Narrow rectangle on lifeline showing when an object is active.
- **Return Message**: Dashed arrow returning control to the caller.

43. What is Cardinality?

Defines the number of instances involved in a relationship.

44. +, -, # Symbols and What They Represent

- +: Public
- -: Private
- #: Protected
  Used in class diagrams to define visibility.

45. Empty Diamond and Filled Diamond Meaning

- **Empty Diamond**: Aggregation (weak relationship)
- **Filled Diamond**: Composition (strong ownership)

46. Why Primary Key Is Not NULL?

Because it uniquely identifies each record, and NULL means unknown or no value.

47. Define Use Case Diagram

A visual tool to depict system functionality, showing actors, use cases, and their interactions within the system boundary.