Integration Testing

Definition:
Integration testing is a type of software testing where we check if different parts (modules) of an application work together correctly.

Why Do We Need Integration Testing?

- To find problems when modules are combined.
- To ensure data is passed between parts without errors.
- To check that the system works as expected when modules interact.

When is Integration Testing Done?

- **After Unit Testing**: (where individual parts are tested alone).
- **Before System Testing**: (where the whole application is tested).

Types of Integration Testing

1. **Top-Down Testing**:
    - Starts from the top module and adds modules one by one downwards.
2. **Bottom-Up Testing**:
    - Begins with the bottom (smallest) module and adds higher-level modules step-by-step.
3. **Big Bang Testing**:
    - All modules are combined and tested together at once. This is used for smaller projects.
4. **Incremental Testing**:
    - Modules are tested in small groups and gradually added to each other.

Common Problems Found in Integration Testing

- **Incorrect Data Flow**: Data is not passed between modules correctly.
- **Interface Errors**: Modules don’t "connect" properly.
- **Unexpected Behavior**: Modules work well alone but fail when combined.

Examples

- Testing a **login** page's connection to a **database**.
- Checking if the **cart** module in a shopping app works with the **payment** system.
- Ensuring the **front end** (what users see) and **back end** (data processing) communicate correctly.

Benefits of Integration Testing

- **Early Problem Detection**: Finds issues in how parts work together before users see them.
- **Reliability**: Makes the application more stable by catching bugs early.
- **Easier Debugging**: Helps find which module is causing the issu