

Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.

Title:

"Software Development Life Cycle (SDLC) - Overview"

Layout Suggestions:

Use a **circular flow chart**, vertical timeline, or step-ladder design to show the progression and interconnection between the phases.

Content for Each Phase:

1. Requirements:

- **Icon:** Clipboard or checklist.
- **Key Message:** "Defining WHAT the system must do."
- **Importance:** Ensures the project meets business needs and user expectations.
- **Outputs:** Software Requirements Specification (SRS), Use Cases.

2. Design:

- **Icon:** Blueprint or UI wireframe.
- **Key Message:** "Mapping out HOW the system will work."
- **Importance:** Translates requirements into architecture and components.
- **Outputs:** System Architecture, Data Models, Prototypes.

3. Implementation:

- **Icon:** Code symbol (e.g., </>).
- **Key Message:** "Turning plans into code."
- **Importance:** Builds the actual product based on design specifications.
- **Outputs:** Source Code, Compiled Applications.

4. Testing:

- **Icon:** Bug symbol or magnifying glass.
- **Key Message:** "Ensuring the system works as intended."
- **Importance:** Detects and fixes defects, ensuring quality and reliability.
- **Outputs:** Test Cases, Bug Reports, Validated Software.

5. Deployment:

- **Icon:** Cloud upload or rocket.

- **Key Message:** "Delivering the software to users."
 - **Importance:** Makes the system available for use and monitors its performance.
 - **Outputs:** Deployed Application, User Manuals.
-

Connecting Arrows/Text:

- Between each phase, add "**Feedback Loops**" to emphasize interconnectivity:
 - E.g., "Revisit Requirements if Testing Reveals Issues."
-

Additional Sections:

- **Title Area:**
Add a **brief SDLC definition**:
"SDLC is a step-by-step framework to develop high-quality software efficiently."
 - **Key Highlights:**
Add "*Why SDLC Matters*":
 - Ensures quality.
 - Minimizes risks.
 - Aligns goals with user needs.
-

Design Tips:

1. Use **vivid colors** to differentiate phases (e.g., blue for Requirements, green for Testing).
2. Include **icons** to make it visually appealing and easy to understand.
3. Keep text concise with bullet points and key phrases.
4. Use tools like **Canva**, **PowerPoint**, or **Figma** for design.