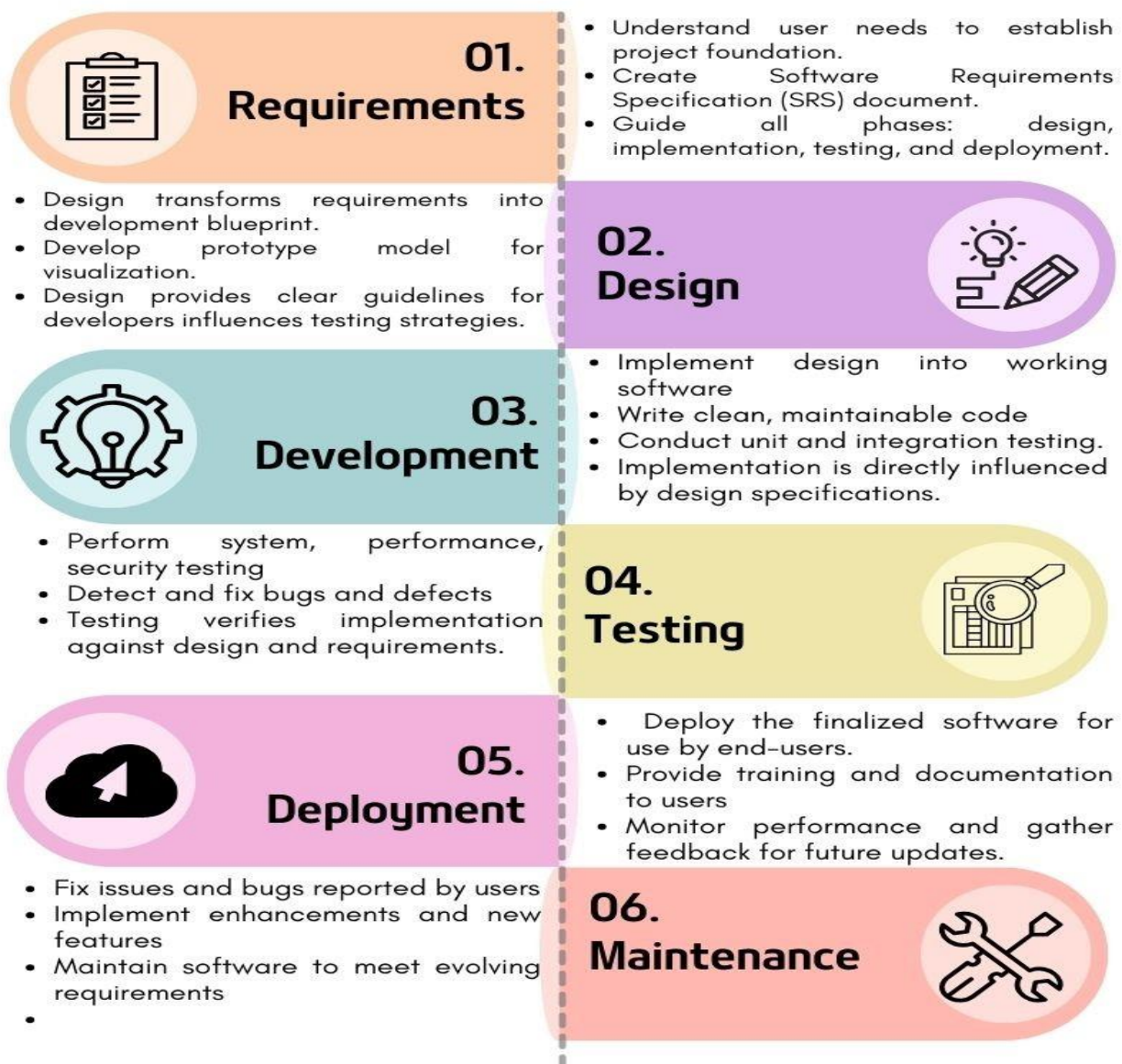# Assignment-1(SDLC)

Create one-page infographic that outlines the SDLC phases (requirements, Design, Implementation, testing, deployment) highlighting the importance of each phase and how they interconnect?

6 STEPS OF SDLC

01. Requirements

- Understand user needs to establish project foundation.
- Create Software Requirements Specification (SRS) document.
- Guide all phases: design, implementation, testing, and deployment.

02. Design

- Design transforms requirements into development blueprint.
- Develop prototype model for visualization.
- Design provides clear guidelines for developers influences testing strategies.

03. Development

- Implement design into working software
- Write clean, maintainable code
- Conduct unit and integration testing.
- Implementation is directly influenced by design specifications.

04. Testing

- Perform system, performance, security testing
- Detect and fix bugs and defects
- Testing verifies implementation against design and requirements.

05. Deployment

- Deploy the finalized software for use by end-users.
- Provide training and documentation to users
- Monitor performance and gather feedback for future updates.

06. Maintenance

- Fix issues and bugs reported by users
- Implement enhancements and new features
- Maintain software to meet evolving requirements

The Software Development Life Cycle (SDLC) consists of different interconnected phases, each phase is important in the development and deployment of high-quality software. Requirements gathering lays the foundation by aligning software with user needs. the design then transforms these requirements into a blueprint, offering clear guidelines for implementation and influencing testing strategies. Implementation translates design specifications into functional code, directly using requirements and design. Testing ensures software quality by validating the software and informing necessary adjustments. Finally, deployment releases the software to users and collects feedback for future updates.

These phases form a continuous process, with each building upon the outcomes of the previous one to deliver software.