Exercise 1: **Describe the stages in the SDLC**

- **Requirement Analysis:**
- Gather and define business requirements.
- Understand user needs and constraints.
- Document functional and non-functional requirements.
- **Design:**
- Create system architecture and design specifications.
- Develop data models and process flows.
- Design user interfaces and user experiences.
- **Development and Testing:**
- Write and compile the source code.
- Perform unit and integration testing.
- Debug and fix code issues.
- **Implementation:**
- Deploy the software to a production environment.
- Conduct system checks and ensure it runs as expected.
- Perform initial user training and support.
- **Documentation:**
- Create user manuals and technical documentation.
- Document system architecture and code.
- Update and maintain documentation as needed.
- **Evaluation:**
- Monitor system performance and user feedback.
- Identify and address any issues or improvements.
- Plan for future updates or enhancements.

Exercise 2: **Compare two software development: the waterfall model and the incremental model**

Waterfall Model:

- **Linear Sequence:**
- Follows a strict, linear progression through phases.
- Each phase must be completed before the next begins.

- **Defined Stages:**
- Distinct stages: Requirements, Design, Implementation, Verification, Maintenance.
- Clear milestones and deliverables at each stage.
- **Documentation Heavy:**
- Emphasis on comprehensive documentation.
- Detailed specifications before development starts.
- **Rigid and Inflexible:**
- Difficult to accommodate changes once a phase is completed.
- Not well-suited for projects with evolving requirements.
- **End Product Delivery:**
- Full product delivered at the end of the cycle.
- Users see the final product only after completion.

Incremental Model:

- **Iterative Development:**
- Develops the software in small, incremental parts (increments).
- Each increment is a functional part of the final system.
- **Flexible and Adaptive:**
- Allows for changes and refinements after each increment.
- Better suited for projects with evolving requirements.
- **User Feedback:**
- Users can provide feedback on each increment.
- Early increments help identify issues and guide further development.
- **Partial Product Delivery:**
- Deliverable product parts after each increment.
- Users see and use parts of the system early in the process.
- **Risk Management:**
- Reduces risks by breaking the project into smaller, manageable parts.
- Problems can be identified and addressed early.