User Requirements

What Are User Requirements?

User requirements are the needs and expectations that users have for a system or product. They describe what the system should do and how it should perform to meet users' needs.

Key Aspects of User Requirements

1. **Functionality**
   - **What it means**: What tasks the system should be able to perform.
   - **Example**: A weather app should show current weather and forecasts.
2. **Performance**
   - **What it means**: How well and how quickly the system should work.
   - **Example**: A website should load within 3 seconds.
3. **Usability**
   - **What it means**: How easy it is for users to use the system.
   - **Example**: A mobile app should have a simple and intuitive interface.
4. **Reliability**
   - **What it means**: How often the system should work correctly without errors.
   - **Example**: An email service should have minimal downtime.
5. **Compatibility**
   - **What it means**: The system should work with other devices or software.
   - **Example**: A software application should run on both Windows and macOS.

Why User Requirements Are Important

- **Meeting Expectations**: Ensures the final product meets what users need and expect.
- **Guidance for Development**: Helps developers understand what features and functions to include.
- **Improves Satisfaction**: Results in a product that users find useful and easy to use.

How to Gather User Requirements

1. **Interviews**: Talk to users to understand their needs.
2. **Surveys**: Ask users to fill out questionnaires.
3. **Observations**: Watch how users interact with current systems.
4. **Feedback**: Collect and analyze user feedback on existing products.

Documenting User Requirements

- **Use Simple Language**: Write in clear, easy-to-understand terms.
- **Be Specific**: Clearly define what each requirement means and how it will be achieved.
- **Organize**: Use headings and bullet points to make the document easy to follow.