**Design Review Checklist**

Description: Use this document as a guide to evaluate the quality, clarity, and structure of the design being reviewed. Enter an X to check each point and leave any sections blank if they do not apply. Be sure to sign your initials at the end of the document once the review is complete.

Name of Reviewer: David Duru

Name of Designer:

**Design Clarity**

|  |  |
| --- | --- |
|  | Design is easy to understand and clearly communicates the intended solution. |
|  | Diagrams and visual aids are clear, complete, and labeled accurately. |
|  | Design choices are justified with appropriate reasoning or references to requirements. |
|  | Any assumptions or constraints are clearly stated and documented. |

**Design Structure and Modularity**

|  |  |
| --- | --- |
|  | System is broken down into clear, well-defined components/modules. |
|  | Components have a single responsibility and are loosely coupled. |
|  | Modules are designed to be reusable and independent. |
|  | The design adheres to SOLID principles (e.g., Single Responsibility, Open-Closed, etc.). |

**Scalability and Performance**

|  |  |
| --- | --- |
|  | The design considers scalability for future growth (e.g., modular architecture, load distribution). |
|  | Potential performance bottlenecks are identified and addressed. |
|  | Design accounts for high availability and fault tolerance. |
|  | Resource usage (memory, CPU) is optimized for the proposed solution. |

**Security Considerations**

|  |  |
| --- | --- |
|  | Sensitive data is handled securely, both in transit and at rest. |
|  | Authentication and authorization mechanisms are included where appropriate. |
|  | Data validation and input sanitization are considered to prevent security vulnerabilities. |
|  | Design includes mechanisms for logging, monitoring, and auditing. |

**Integration and Compatibility**

|  |  |
| --- | --- |
|  | Interfaces between modules are clearly defined and documented. |
|  | External dependencies are clearly specified and justified. |
|  | The design considers compatibility with existing systems or components. |
|  | Integration points are identified, and data flow between components is clearly illustrated. |

**User Experience (UX) and Usability**

|  |  |
| --- | --- |
|  | User interfaces are intuitive and designed with the end-user in mind. |
|  | The design follows UX best practices and accessibility standards. |
|  | User journeys and flows are clear and easy to follow. |
|  | Appropriate feedback mechanisms (e.g., error messages, confirmations) are included. |

**Maintainability and Extensibility**

|  |  |
| --- | --- |
|  | Design is easy to maintain and extend in the future. |
|  | Changes to one module do not require extensive changes in other modules. |
|  | Design is well-documented to facilitate future maintenance and updates. |
|  | Naming conventions and design patterns are consistent throughout. |

**Documentation**

|  |  |
| --- | --- |
|  | Diagrams (e.g., ER diagrams, flowcharts) are up-to-date and relevant. |
|  | Design rationale and trade-offs are documented. |
|  | Dependencies, configurations, and environment setups are detailed. |
|  | Any APIs or third-party services used are clearly referenced and explained. |

**Consistency with Requirements**

|  |  |
| --- | --- |
|  | Design meets all functional and non-functional requirements. |
|  | Each requirement is traced back to its implementation in the design. |
|  | Any changes to requirements are documented and reflected in the design. |

Reviewer Initials: