Raspberry Pi System Monitor

| **Criteria** | **Exemplary (4)** | **Proficient (3)** | **Basic (2)** | **Limited (1)** |
| --- | --- | --- | --- | --- |
| **Software** | Exceptional understanding of software concepts, showcasing depth and breadth of knowledge. Implements complex functionalities with efficiency and elegance. Demonstrates advanced skills in selecting and integrating diverse libraries or tools. | Solid understanding of software concepts and their application to the project. Effectively uses programming/scripting languages to implement required functionalities. Demonstrates proficiency in selecting and utilizing relevant libraries or tools. | Adequate understanding of software concepts relevant to the project. Uses programming/scripting languages appropriately, but with some gaps in implementation. Demonstrates basic utilization of libraries or tools, with room for improvement. | Limited understanding of software concepts and their relevance to the project. Minimal use of relevant programming/scripting languages. Demonstrates difficulty in identifying and utilizing appropriate libraries or tools. |
| **Computer Interfacing** | Exceptional understanding of computer interfacing concepts, showcasing depth and breadth of knowledge. Implements advanced communication protocols and techniques. Demonstrates mastery in configuring and managing diverse hardware interfaces. | Solid understanding of computer interfacing concepts and their application. Establishes reliable communication between components or devices. Manages hardware interfaces effectively, demonstrating proficiency in configuration and troubleshooting. | Demonstrates basic understanding of computer interfacing concepts. Establishes communication between components or devices with occasional issues. Manages hardware interfaces adequately, but with room for improvement. | Limited understanding of computer interfacing concepts and their application. Struggles to establish basic communication between components or devices. Shows difficulty in configuring and managing hardware interfaces. |
| **Internet of Things** | Exceptional understanding of IoT concepts, showcasing depth and breadth of knowledge. Integrates and optimizes IoT devices with advanced features. Demonstrates mastery in designing and implementing complex IoT data communication systems. | Solid understanding of IoT concepts and their relevance to the project. Integrates IoT devices effectively. Demonstrates proficiency in managing data communication in an IoT context. | Demonstrates a basic understanding of IoT concepts. Integrates IoT devices with occasional issues. Shows competence in basic data communication within an IoT framework. | Limited understanding of IoT concepts and their application to the project. Struggles to integrate IoT devices or lacks awareness of IoT principles. Demonstrates difficulty in handling data communication in an IoT context. |
| **Troubleshooting** | Effectively identifies and resolves issues, implementing comprehensive error handling mechanisms, and providing detailed troubleshooting guidelines. | Demonstrates the ability to troubleshoot common issues, with minor gaps or oversights in error handling and troubleshooting documentation. | Shows a foundational understanding of troubleshooting, but encounters notable challenges or limitations in identifying and resolving issues. | Struggles significantly with troubleshooting, resulting in persistent issues and a lack of effective error handling. |

| **Criteria** | **Exemplary (4)** | **Proficient (3)** | **Basic (2)** | **Limited (1)** |
| --- | --- | --- | --- | --- |
| **Define Program Function** | The program function is well-defined, clear, and comprehensive. It includes all key features necessary for a recipe app, and the scope is appropriately defined, taking into account the target audience and user needs. | The program function is mostly clear and includes most of the essential features for a recipe app. Some aspects may require further refinement. | The program function is partially defined, but it lacks clarity or completeness. Key features may be missing or vaguely described. | The program function is poorly defined and lacks clarity. It is missing essential features or does not consider the needs of the target audience. |
| **Program Design** | The wireframes and mockups are highly detailed, visually appealing, and effectively communicate the app's design. The design aligns with the defined program function and user needs. | The wireframes and mockups are well-crafted and provide a clear representation of the app's design. They generally align with the defined program function and user needs. | The wireframes and mockups are somewhat detailed but may lack consistency or clarity in design. They require further refinement to fully align with the program function. | The wireframes and mockups lack detail, consistency, or clarity, making it challenging to understand the app's design. Significant improvements are needed. |
| **Identifying and Correcting Errors** | Errors or design flaws are proactively identified and thoroughly corrected. The design demonstrates a high level of attention to detail and usability. | Most errors or design flaws are identified and corrected during the design process. The design shows good attention to detail and usability. | Some errors or design flaws are identified and corrected, but others remain unresolved. The design may lack consistency or exhibit minor usability issues. | Few errors or design flaws are addressed, and many issues remain unresolved. The design lacks attention to detail and usability. |
| **Working in Teams** | Team members collaborate effectively, with clear roles and responsibilities. Communication is excellent, and the project is well-organized and on schedule. | Team members work well together, with defined roles and good communication. The project is generally organized, with minor deviations from the schedule. | Team members work together but may struggle with role definition and communication. The project has some organizational challenges and deviations from the schedule. | Team members have difficulty working together, with unclear roles and poor communication. The project is disorganized, and there are significant deviations from the schedule. |
| **Development Cycle** | The project follows a well-structured development cycle with clear phases, milestones, and deadlines. Each phase is executed meticulously, and the team consistently meets or exceeds the schedule. | The project follows an organized development cycle with defined phases and milestones. Most deadlines are met, and the project generally progresses smoothly. | The development cycle is somewhat structured, but there may be some deviations from the plan. Some milestones and deadlines are met, but the project may experience delays or challenges. | The development cycle lacks clear structure and organization. Milestones and deadlines are frequently missed, leading to significant project delays. |
| **Documentation** | The project documentation is comprehensive, well-organized, and clearly articulates the design rationale, user feedback, and revisions made. It serves as an excellent resource for future reference. | The project documentation is well-structured and covers essential aspects of the design process, including design rationale, user feedback, and revisions. It is a valuable resource for reference. | The project documentation is somewhat organized but may lack completeness or clarity in certain areas. It provides a basic overview of the design process. | The project documentation lacks organization, and key aspects of the design process, such as design rationale, user feedback, and revisions, are missing or unclear. |
| **User Testing** | User testing is conducted with a diverse group of target users, and feedback is extensive and insightful. The app design is refined based on user feedback, resulting in a highly user-friendly design. | User testing is conducted with relevant target users, and feedback is generally constructive. The app design is refined based on user feedback, enhancing usability. | User testing is conducted with a limited group of users, and feedback may lack depth or diversity. The app design is adjusted based on some user feedback but may not fully address all usability issues. | User testing is minimal or nonexistent, and little to no user feedback is incorporated into the app design. Usability issues may remain unaddressed. |