Raspberry Pi System Monitor Project Guide

**Project Description:** Create a bash script for a Raspberry Pi that functions as a system monitor. The script will gather information about the Raspberry Pi's operating system, file system, software, and basic Linux commands. It will present the information in a user-friendly format and include the ability to launch web browsers or applications.

# Phase 1: Planning

### Project Objective

* + Clearly define the goal of the project.

|  |
| --- |

### Hardware Requirements

* + Identify the necessary hardware components, including sensors and peripherals.

|  |
| --- |

### Visualization

* + Identify the key features and functionalities of the system monitor script.

|  |
| --- |

### Project Scope

* + Determine the Linux commands and utilities to be used for gathering system information.

|  |
| --- |

* + Specify the structure of the menu-driven interface.

|  |
| --- |

### Timeline

* + Create a timeline that outlines the estimated duration for each phase of the project.

| **Task** | **Who** | **Start** | **End** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Resource Requirements

* + Identify the tools and libraries needed.

|  |
| --- |

# 

# Phase 2: Design

### User Interface Design

* + Create a wireframe or mockup of the user interface using a design tool or on paper.

|  |
| --- |

### Algorithm Design

* + Define the structure and logic of the application, including how user input will affect the program.

|  |
| --- |

### Data Abstraction

* + Develop the procedures/functions.

|  |
| --- |

### Flowchart / Pseudocode

* + Generate a flow chart or pseudo code of this application.

|  |
| --- |

### Error Handling

* + Plan how you will handle potential errors.

|  |
| --- |

### Test Cases

* + Develop a set of test cases that will be used during the testing phase to ensure the app works as expected.

|  |
| --- |

# 

# Phase 3: Testing and Feedback

### Unit Testing

* + Test individual components of the weather station app, such as sensor connection, sensor data, input validation, and error handling.

|  |
| --- |

### Integration Testing

* + Verify that all components work together cohesively within the app.

|  |
| --- |

### User Testing

* + Invite potential users to test the app and provide feedback on its usability and functionality.

|  |
| --- |

### Bug Fixing

* + Address any issues or bugs identified during testing.

|  |
| --- |

# 

# 

# Phase 4: Documentation

### Hardware Schematic

* + Create a schematic for the design of the hardware.

|  |
| --- |

### User Manual

* + Create a user manual that explains how to use the calculator app, including instructions on entering numbers, selecting operations, and interpreting the results.

|  |
| --- |

### Code Documentation

* + Document the code, including comments that explain the purpose of each function and segment of code.

|  |
| --- |

### Project Report

* + Write a comprehensive project report that summarizes the planning, design, and testing phases, as well as the skills and objectives achieved in the project.

|  |
| --- |

# 

# Reflection

**1. Networking Concepts:**

* What were the key networking concepts you encountered and applied in the project?
* How did your understanding of networking concepts evolve throughout the project?
* What challenges did you face in implementing the peer-to-peer network, and how did you overcome them?

**2. Basic Networking Security:**

* Describe the basic networking security measures implemented in the project. How effective were they?
* What security challenges did you encounter, and how did you address them?
* Reflect on the importance of basic networking security in the context of your project.

**3. IoT Integration:**

* Discuss the process of integrating IoT devices into the Raspberry Pi network. What sensors did you use, and how did they contribute to the project?
* Reflect on the role of IoT in enhancing the functionality of your peer-to-peer network.
* What were the main challenges and successes in integrating IoT devices, and how did they impact the overall project?

**4. Troubleshooting:**

* Describe the troubleshooting mechanisms you implemented. How effective were they in identifying and resolving issues?
* What were the most common issues encountered during the project, and how did you approach their resolution?
* Reflect on the importance of having robust troubleshooting mechanisms in a complex project like this.