Shelly Christmas Coding Challenge

Individual Project Assignment

This is the Individual Project Assignment for the Shelly Christmas Coding Challenge

1. General Requirements

Your project should meet the following requirements and development techniques.

1. **Documentation with Pictures**: *(0 – 10 points)* The project should be well-documented, including clear and concise explanations of the code and functionalities. Visual aids, such as pictures, diagrams, or flowcharts, should be included to enhance understanding and improve the overall clarity of the project. This helps the developer or viewer understand how to interact with the code after a long time without seeing it or for the first time.

2. **Error Handling:** *(0 – 6 points)* The code should incorporate error handling
    - **Exception Handling** *(3 points)***:** Implement a mechanism to catch and handle exceptions that may occur during code execution.
    - **User-Friendly Messages** *(3 points)*: Provide clear and meaningful error messages that help users understand what went wrong and how to address the issue.

3. **Validation:** (0 - 4 points)
    - **Input Validation:** Validate data to ensure they meet specified criteria (e.g., data type, length, format) before processing.

4. **Integration with External Services**: *(0 – 20 points)*
    - **API Compatibility:**
        - The project should be capable of seamlessly integrating with external APIs. *(10 points)*
        - It should support common API communication methods such as RESTful. *(0 – 10 points)*
            - **Request Handling:** Develop the ability to send requests to external services for data retrieval or action execution. *(5 points)*
            - **Response Processing:** Implement mechanisms to parse and handle responses received from external services. *(5 points)*

5. **Well-Written Code**: *(0 – 20 points)* The project's code should follow best practices, demonstrating clean, readable, and maintainable code.
    - **Readability:** *(0 - 9 points)*
        - The code should be well-organized and easy to follow. *(3 points)*
        - Variable and function names should be descriptive. *(3 points)*
        - Proper indentation and formatting. *(3 points)*
    - **Modularity:** *(0 - 9 points)*
        - Code modularity. *(3 points)*
        - Variables, methods and/or classes should be reused where it needs *(3 points)*
        - **Naming:** *(0 - 3 points)*
            - **Variables** – Variable names should be related to what value they hold. *(1 point)*
            - **Functions** – Function names should be related to what logic it performs. *(1 point)*
            - **Classes** – Class names should be related to the objects they will cover. *(1 point)*
    - **Comments:** *(0 – 2 points)*
        - Sufficient comments to explain complex sections of code. *(2 points)*

6. **GitHub Publication**: *(0 – 10 points)* The project should be published on GitHub, ensuring easy accessibility for review and evaluation.
    - Participants should use branching model (creating new branch for every problem) of git *(6 points)*
    - Good, short and describable commit messages *(2 points)*
    - Keeping track of code changes effectively *(2 points)*

2. Additional Requirements

7. **Reproducible Results Instructions**: *(0 – 5 points)* The README file should provide comprehensive instructions on how to reproduce the project's results. This should include step-by-step guides, dependencies, installation procedures, and any other necessary configurations to ensure smooth execution in different environments

8. **Event Handling:** *(0 – 10 points)* Implement event handling in your script using Shelly's **addEventHandler()** and **addStatusHandler()** methods. These enable dynamic responses to internal events. Explore custom event creation with **emitEvent()** to broadcast events across RPC channels.
    - Manage subscriptions effectively, removing listeners when necessary using appropriate methods. This requirement encourages a responsive and modular script design based on event-driven programming paradigms

9. **Data Storing:** *(0 – 15 points)* Saving data in DB is the best way to access them later even if the device isn't working.
    - **KVS (Key-Value Store)** – KVS is the best way, but it has some limits
    - **NOSQL or SQL DB –** Using one of these may be challenging, but if you have lots of data this is the way

3. Assessment Criteria

General Requirements – 70%

- **Documentation with Pictures** – **0…10**
- **Error Handling** – **0…6**
- **Validation – 0…4**
- **Integration with External Services** – **0…20**
- **Well-Written Code** – **0…20**
- **GitHub Publication** – **0…10**

Additional Requirements – 30%

- **Reproducible Results Instructions** – **0…5**
- **Event Handling – 0…10**
- **Data Storing – 0…15**

Bonuses – up to 15%

10. **Multiple Devices Usage**: The project could demonstrate the ability to work with more than one device. This could involve integrating various Shelly devices or other IoT devices to showcase the project's scalability and adaptability in a real-world environment
11. **Web Interface**: As a bonus requirement, participants can create a web-based user interface for their IoT project. The web interface should allow users to interact with the IoT devices, view real-time data, and control various functionalities remotely
12. **Advanced Functionalities**: The project could utilize functionalities not explicitly covered in the course (such as **MQTT**, **Schedule**), showcasing the participants' creativity and ability to explore advanced programming concepts. This could involve implementing automation scenarios or incorporating additional features to enhance the overall functionality of the project

3. Submission Deadline

The final date for sending the GitHub link to your project is **31 December, 11:00 PM (CET)**.

4. Challenge Rewards

The prize categories for the **best three projects** are:

- **First Place Prize**: 500 € voucher for Shelly Products
- **Second Place Prize**: 300 € voucher for Shelly Products
- **Thirds Place Prize**: 100 € voucher for Shelly Products

The ranking of the projects is done **based only on the submitted project**.