Project guideline
The development of a project is OPTIONAL to overcome the exam It is ok to develop a project that is not present in this list, after written agreement by the Professor Multiple choice question and project discussion can occur in separate exam sessions A project can be chosen by a group composed of maximum 2 persons Each project must be original! Each project chosen must be communicated to the Professor via email, together with a GitHub account username Students will receive a GitHub link with a repository where to upload the source code and all the project materials MAXIMUM 2 days before the exam An email must be sent to the Professor after the project upload on GitHub is completed
Integration of MQTT Server and Python for Big Data Analysis and Storage in Different Database Platforms Description of the project: The project aims to develop a system capable of collecting, processing and storing large volumes of data from sensors or IoT devices using the MQTT (Message Queuing Telemetry Transport) protocol and Python. The system will be able to manage the reception of data via MQTT server, analyze it in real time and store it in different types of databases, including SQL, MongoDB and Neo4j, based on the topic of the MQTT messages. Project Track: 1. Introduction: - Problem description: The growing adoption of IoT devices has led to an exponential increase in the amount of data generated, making it essential to develop efficient systems for managing and analyzing this big data. - Project objective: Create an integrated system that uses MQTT server and Python to receive, process and store data from IoT sensors in different types of databases. 2. Implementation of MQTT Server: - Selecting an MQTT server: Review the different options available (e.g. Eclipse Mosquitto, EMQ X, etc.) and choose the one that best suits your project needs. - Server configuration: Installation and configuration of the MQTT server to allow the reception and forwarding of MQTT messages from IoT devices. 3. Python Application Development: - Using the Paho MQTT library: Use the Paho MQTT library to develop a Python application capable of connecting to the MQTT server and subscribing to specific topics for receiving data. - Data Processing: Implement logic for processing received data, including transforming MQTT messages into a format usable for analysis. 4. Database Integration: - Database selection: Examine the characteristics and requirements of different database types (SQL, MongoDB, Neo4j) and choose the most appropriate approaches for storing IoT data. - Connection to databases: Implement connections to different database platforms using specific libraries for each (for example, pymongo for MongoDB, py2neo for Neo4j). - Data writing: Define the logic to write the processed data to the respective databases based on the topic of the MQTT messages. For example, temperature data could be stored in a SQL database, while device network data could be stored in a graph database like Neo4j. 5. Test and Evaluation: - Functional tests: Verify the correct functioning of the system through a series of tests that include data reception, processing and storage in the various databases. - Performance evaluation: Measure system performance in terms of data processing speed, scalability and reliability. 6. Conclusions: - Summary of results: Summarize the results obtained during the development and testing of the system. - Possible future developments: Identify any areas for improvement or extension of the system and suggest possible future developments. 7. Documentation and Presentation: - Technical documentation: Create detailed documentation that illustrates the system architecture, the technologies used and instructions for installation and use. - Project presentation: Prepare a presentation illustrating the project, its objectives, implementation and results obtained to share with faculty and colleagues.

Id	Title	Description
DB-B1	Environmental Monitoring	Using or simulating IoT sensors to collect data on air quality, temperature, and humidity
DB-B2	Fleet Tracking	Utilizing or simulating IoT devices installed on vehicles to gather data on location, speed, and vehicle conditions
DB-B3	Industrial Monitoring	Using or simulating IoT sensors to monitor operational parameters of machines in an industrial plant

DB-B4	Precision Agriculture	Employing or simulating IoT sensors to collect data on soil moisture, temperature, and light in agricultural crops
DB-B5	Urban Surveillance	Using or simulating IoT cameras to collect real-time video data from various urban areas
DB-B6	Energy Resource Management	Using or simulating IoT sensors to monitor energy consumption in residential and commercial buildings
DB-B7	Home Security	Using or simulating IoT sensors to detect intrusions, fires, or gas leaks in homes
DB-B8	Health and Wellness	Using or simulating IoT wearable devices to monitor vital signs such as heart rate and level of physical activity
DB-B9	Traffic Management	Using or simulating IoT sensors to monitor vehicle and pedestrian flow in urban areas
DB-B10	Social Media Analysis	Using APIs to collect data from social media on conversation trends, user sentiment, and influencers