**Final Group Project**

**AI – IOT**

**VT 25**

**Group Project Instructions**

**1. Project Scope and Purpose**

The final course project challenges your group to design, build, and present an **AI- and IoT-driven application** that incorporates **AWS Rekognition**, **AWS Lex**, and **real-world or mock IoT data** (e.g., weather, environmental sensors, etc.).

This project brings together the concepts you’ve learned throughout the course—Machine Learning, IoT, cloud integration, and conversational interfaces—into a practical, problem-solving application.

You are encouraged to think big but start small. A well-executed minimal version (**MVP**) with clear logic and working AWS integration is better than an overambitious, incomplete prototype.

**2. Project Foundation Options**

Each group can choose one of the following approaches for your project:

* Use **one team member’s previous draft or prototype** as the foundation and iterate/build on top of it.
* **Combine ideas** from multiple members to form a cohesive, collaborative application.
* Start with a **fresh new idea** that the group collectively agrees on.

**3. Minimum Required Components**

Your project **must** include the following components:

**AWS Rekognition**

* For **image-based analysis**, such as detecting plant diseases, object identification, facial analysis, etc.

**AWS Lex**

* A **chatbot interface** that allows users to interact with your system, ask questions, or request results (e.g., "What's the weather forecast?", "What disease is on my tomato plant?", etc.).

**IoT Data Integration**

* Integrate at least one **IoT-like data source**, which can be:
  + Real APIs (e.g., **SMHI**, **OpenWeatherMap**, etc.)
  + Simulated/mock sensor data (e.g., soil moisture, temperature, pH levels)

**4. Optional Enhancements**

While **not required**, you are encouraged to enhance your project with the following:

* **User Interface (UI)**:
  + Simple interfaces using tools like **Streamlit**, **Flask**, or even CLI interfaces to make the app more user-friendly.
  + A web-based frontend is **not mandatory**, but some way of interacting with the system is expected.
* **Lambda Functions**:
  + Use **AWS Lambda** to glue together Rekognition, Lex, and IoT APIs into a serverless pipeline.
* **Cloud Storage**:
  + Store image uploads or other artifacts using **AWS S3**.
* **Machine Learning Model**:
  + You may include a **custom ML model** trained on your dataset if you want to go beyond Rekognition.

**5. Deliverables**

Each group should submit the following:

1. **Project Report (PDF)** – Include:
   * Project overview
   * Architecture diagram
   * AWS services used and why
   * Description of IoT data and how it’s used
   * Screenshots of chatbot/UI in use
   * Challenges faced and how you solved them
   * Roles and responsibilities within your group
2. **Code Repository** – Hosted on GitHub or similar, including:
   * Source code
   * Setup instructions
   * Any mock data or config files required to run
3. **Final Presentation** – A 20–30 minute in-class or online presentation showcasing your working application and architecture.