



AVICO

CODING TRAINING WITH AVIATION TECHNOLOGIES

Methodological framework for vocational education and
training providers



Co-funded by
the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Contents

Introduction	3
1. AVICO MOOC Overview	4
2. MOOC Platform User Guide	5
2.1. Accessing the Platform.....	5
2.1.1. Access via a local user account	6
2.2. Course Overview.....	8
2.3. Communication and Support	11
3. Experiment Kit (Simulation) Guide	12
3.1. Purpose and role of the experiment Kit.....	12
3.2. Pedagogical use scenario	13
3.3. Technical specifications	14
3.3.1. Software requirements	15
3.3.2. Hardware components	17
3.4. Structure of experimental tasks	19
3.5. Practical simulation scenarios	21
3.6. Assessment and evaluation.....	23
3.6.1. Scenario-specific assessment rubrics	23
3.7. Teacher's manual and facilitation	24
3.7.1. Facilitation strategies	24
3.7.2. Troubleshooting and common errors	25
3.8. Annex: Ready-to-use materials	26
3.8.1. Checklist template: Pre-flight safety	26
3.8.2. Observation sheet: data analysis	26
4. Adoption and replication framework	27
4.1 Purpose and strategic approach.....	27
4.2. Adoption process and integration into VET practice.....	27
4.3. Adaptation to learners' needs and implementation models.....	28
5. Sustainability and long-term use	28



Introduction

The AVICO project – Coding Training with Aviation Technologies is an Erasmus+ Cooperation Partnership in vocational education and training that aims to strengthen the connection between digital skills development, coding education, and unmanned aerial vehicle (UAV) technologies. The project responds to the rapid expansion of UAV applications across sectors such as agriculture, logistics, environmental monitoring, media, and public services, while recognising that vocational education systems often lack structured learning pathways that effectively combine drone technologies with coding and computational thinking.

A central outcome of the project is the development of a dedicated Moodle-based learning environment that hosts the AVICO training programme. This platform integrates curriculum content, open educational resources, coding exercises, UAV-related materials, and simulation-based activities within a structured and interactive online learning space. Designed to be accessible and adaptable, the platform supports vocational learners, educators, and training providers in engaging with innovative and practice-oriented learning experiences.

By combining coding as a transversal digital competence with UAV technologies as an applied learning context, AVICO promotes the development of technical skills, analytical thinking, and problem-solving abilities. At the same time, the Moodle Learning Management System (LMS) enables flexible and collaborative learning, allowing users to access materials, participate in activities, monitor progress, and interact with digital tools aligned with the project's learning objectives.

Within this framework, this document has been developed to support Vocational Education and Training (VET) providers in effectively understanding, adopting, and implementing the AVICO MOOC platform and simulation-based experiment kit. The guide goes beyond a purely technical explanation of the tools by providing a comprehensive methodological framework that enables teachers, trainers, and institutions to respond to the evolving needs of their students and trainees, particularly in relation to digital transformation, STEM education, and emerging technologies.

In a rapidly changing labour market, VET providers are increasingly required to equip learners with practical, transferable, and future-oriented skills. This document therefore combines technical guidance, pedagogical principles, and dissemination strategies into a single, coherent resource. It is designed to support VET providers in understanding the structure and educational value of the AVICO tools, applying them in diverse teaching and training contexts, and adapting them to different learner profiles, institutional capacities, and local needs.

As such, the document functions both as a practical user guide for navigating and using the AVICO learning environment, and as a strategic implementation framework that facilitates the transfer, scalability, and sustainability of project results across different VET systems in Europe.



1. AVICO MOOC Overview

The AVICO MOOC (Massive Open Online Course) is an integral component of the AVICO project, which aims to modernize vocational education by integrating coding education and unmanned aerial vehicle (UAV) technologies into teaching and learning processes. The platform is designed as an open-access, digital learning environment that supports flexible, scalable, and innovative education for a broad range of learners, particularly within vocational education and training (VET) contexts.

Objectives of the MOOC

The main objectives of the AVICO MOOC are to:

- Enhance vocational education through the integration of digital technologies and coding skills
- Introduce learners to UAV (drone) technologies as an applied learning context
- Promote project-based and practice-oriented learning approaches
- Support educators with open educational resources (OER) and structured curricula
- Improve digital competences and employability skills of learners in a modern labour market

Overall, the MOOC aims to bridge the gap between traditional vocational training and the demands of a rapidly evolving digital and technological environment.

Structure of the MOOC

The AVICO MOOC is structured as a modular online learning system that allows learners to progress through content in a flexible and self-paced manner. Key structural elements include:

- Thematic modules focused on coding and UAV applications
- Video lectures and instructional materials
- Exercises and practical coding tasks
- Simulation-based learning activities using drone-related scenarios
- Quizzes and self-assessment tools
- Multilingual course availability to support accessibility across partner countries

The structure follows a progressive learning model, where theoretical knowledge is continuously connected with practical application.

Learning Outcomes

Upon completion of the AVICO MOOC, learners are expected to:

- Understand basic and intermediate coding principles
- Apply coding skills in UAV and simulation environments



- Develop problem-solving and computational thinking skills
- Work effectively in project-based learning settings
- Demonstrate improved digital literacy and technical competence
- Apply knowledge in real-world vocational and technological contexts

These outcomes are designed to support both educational development and future employability in technology-oriented sectors.

Pedagogical logic

The pedagogical foundation of the AVICO MOOC is based on modern learner-centred approaches, combining constructivist and experiential learning principles. The core pedagogical logic includes:

- **Project-based learning (PBL):** learners acquire knowledge through real-world tasks and UAV-related projects
- **Blended coding education:** combining block-based and text-based programming approaches
- **Experiential learning:** emphasis on learning by doing through simulations and practical exercises
- **Self-paced learning:** allowing flexibility and adaptation to individual learning needs
- **Collaborative learning environment:** encouraging interaction among learners and educators across countries

This approach aligns with contemporary MOOC design principles, which emphasize scalability, accessibility, and interactive engagement in digital learning environments.

2. MOOC Platform User Guide

2.1. Accessing the Platform

Welcome to the AVICO e-learning platform! This system is hosted on <https://mooc.avico-project.com/?lang=en> platform, which is specifically designed to support interactive learning, professional development, and cross-institutional collaboration within educational projects. Access to the platform is available in this way. For all the participants, access is through email registration: fill out the form, verify your email, and log in with the credentials you create.

This login options grant complete access to all AVICO course materials, digital resources, and platform features.

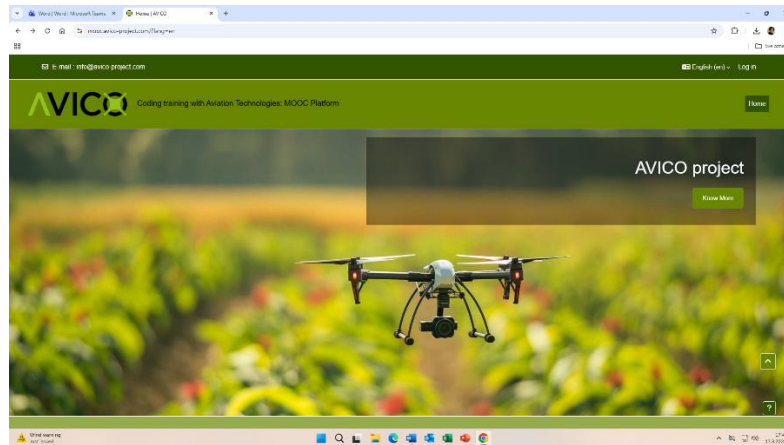


Figure 1. AVICO dashboard

2.1.1. Access via a local user account

1. Visit the Platform: Open your web browser and go to:
<https://mooc.avico-project.com/login/index.php>
2. Select Log in using a local user account

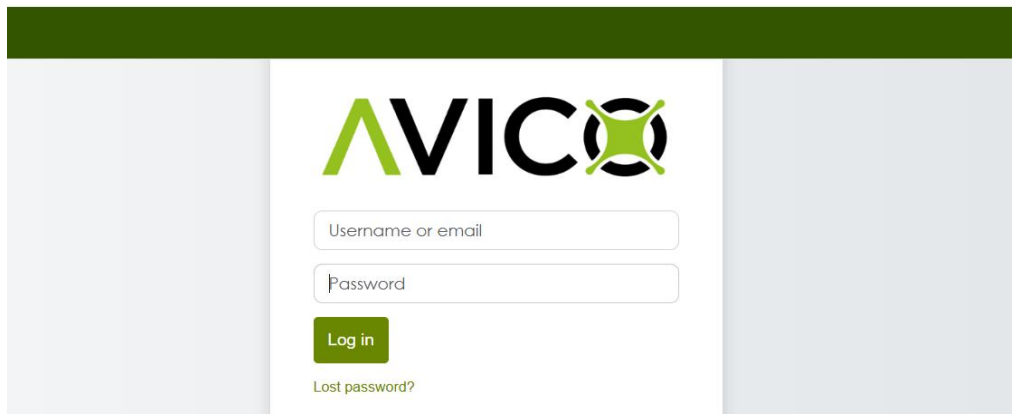


Figure 2. Log in using a local user account

3. If you do not have an account, select the option to **Create a new account.**

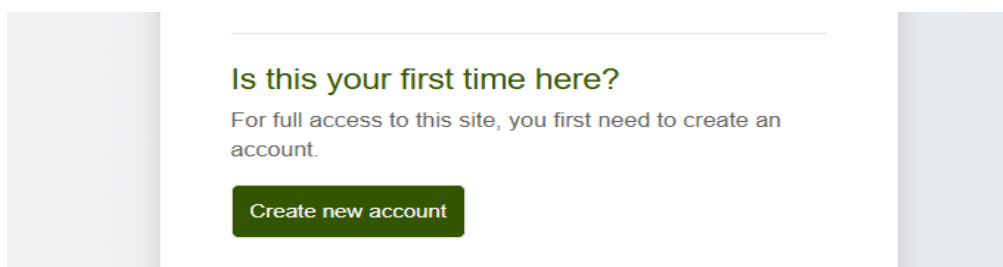


Figure 3. Creation of a new account

Is this your first time here?

You must complete the following steps in order to gain access to e-courses:

1. Fill out the [New Account](#) web form with your personal details.
2. After submitting the form, you should receive an email at the email address you provided.
3. Please carefully read the email and click on the link in order to confirm your registration and log in to the system.

Check out our [quick guide](#) on logging into the MoD system.

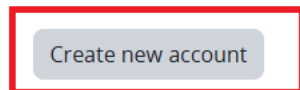
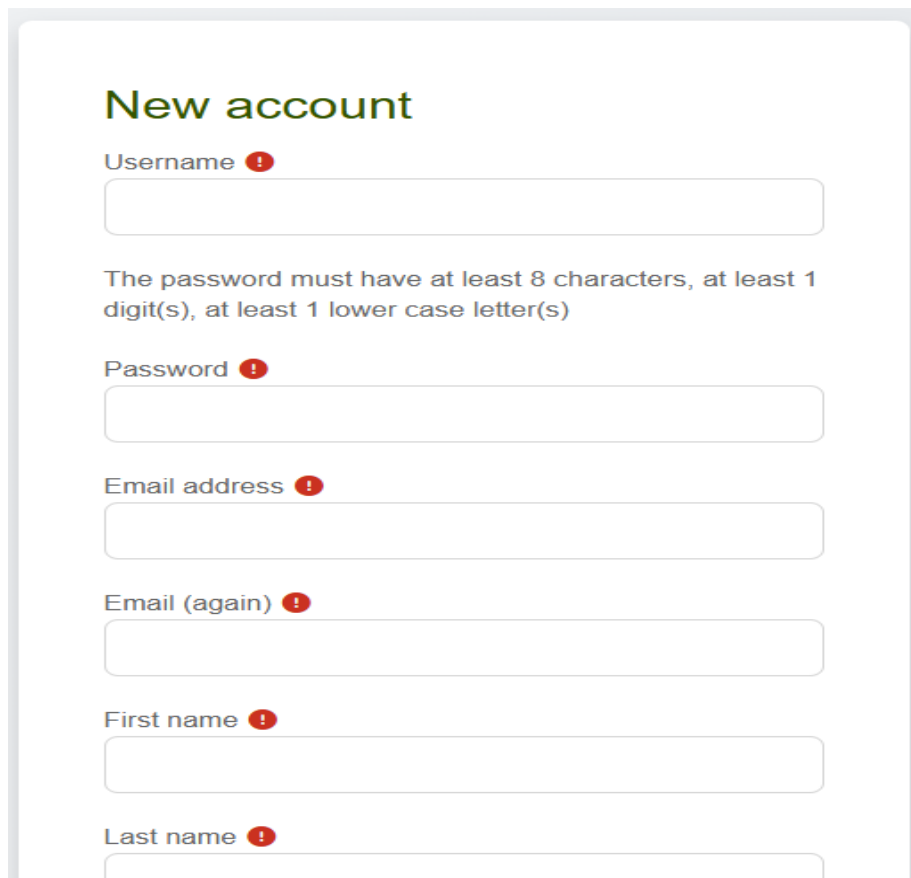



Figure 4. Create a new account

Select the Option to Register with Your Email Address


Fill out the registration form by entering the required personal details (name, email address, password, city, country etc.).





New account


Username 

The password must have at least 8 characters, at least 1 digit(s), at least 1 lower case letter(s)

Password 

Email address 

Email (again) 

First name 


Last name 

Figure 5. Registration form

Click **Create my new account** to complete the registration.



Verify Your Email

After submitting the form, you will receive an email with a confirmation link.

An email should have been sent to your address at .@gmail.com

It contains easy instructions to complete your registration.

If you continue to have difficulty, contact the site administrator.



Figure 6. Check an email with a confirmation link.

Click on the link in the email to verify and activate your account.

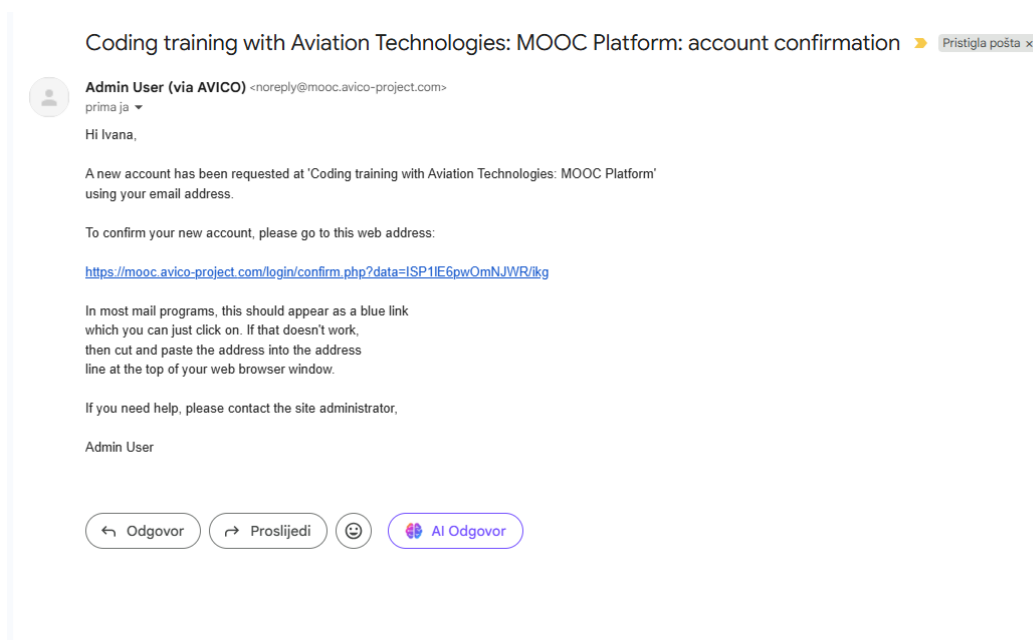


Figure 7. Verification and activation

2.2. Course Overview

This section offers a complete guide on how to access and navigate your courses effectively. From locating your enrolled courses to exploring individual topics and finishing assignments, you will understand the platform's layout and features.

Access Course List

Once you log in, you will find a navigation menu in the top left corner with the following options: **Home**, **Dashboard**, and **My courses**. Click on **My courses** to view the list of courses you are enrolled in.



Figure 8. Navigation menu

Click on the course name to access its content. You are choosing between the enrollment of

courses in the consortium languages.

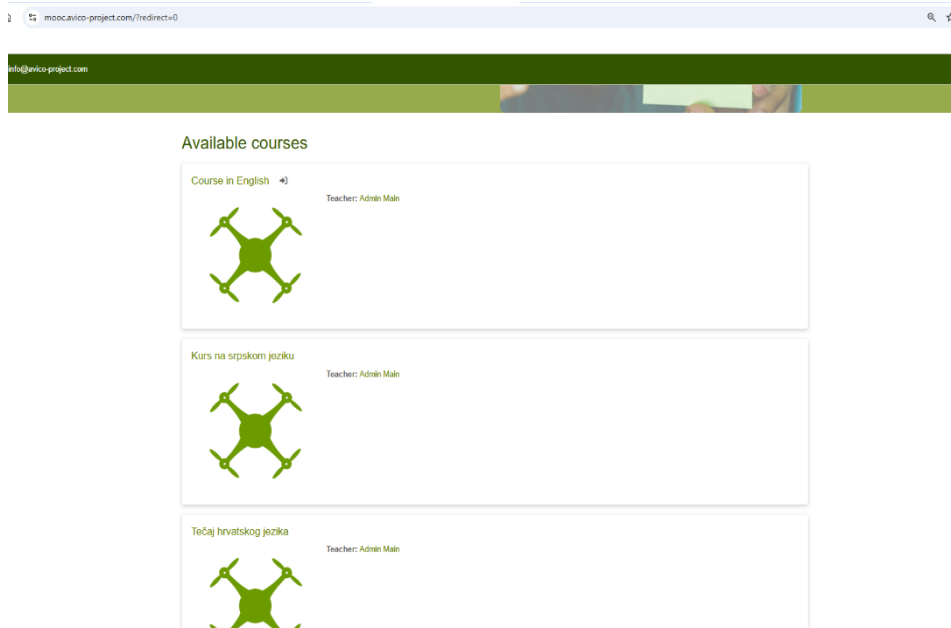


Figure 9. Available courses

No enrollment key is required, participants can self-enroll.

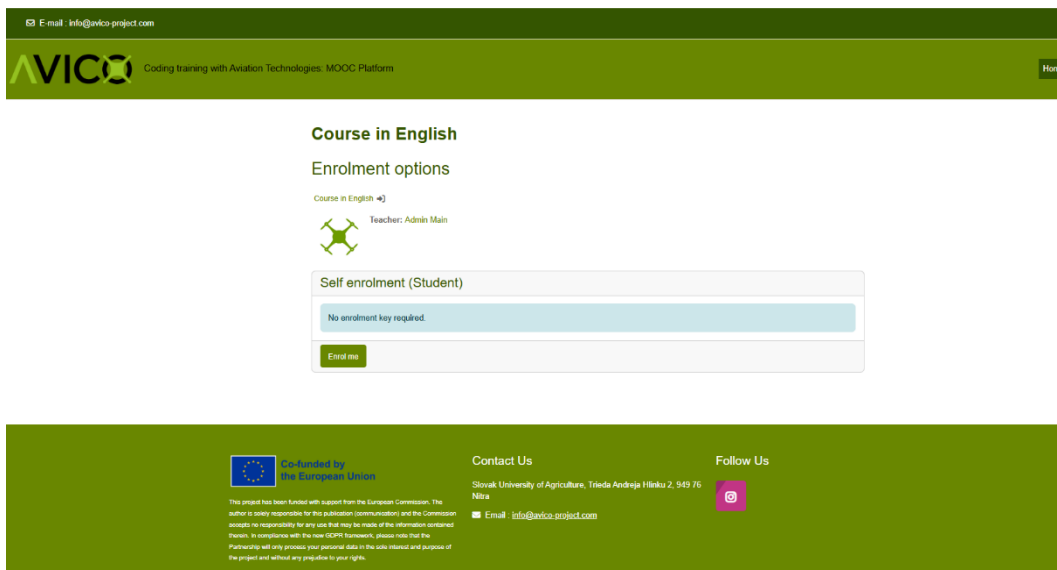
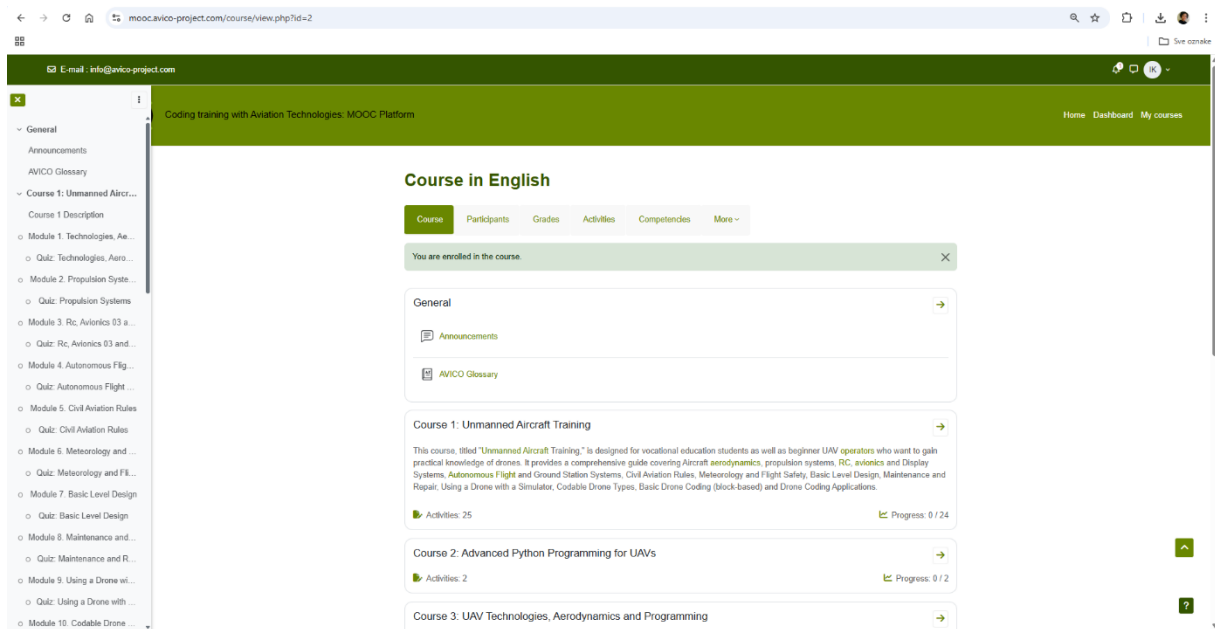


Figure 10. Enrollment into courses

Navigating the Course

The **General topic** acts as the main hub for the course. It offers key information like a brief overview of the course objectives and structure, along with important announcements or updates related to the course (**Notifications**). **Also, this section presents the Glossary section of the AVICO platform.**

The course content is arranged into topics on the left side. Click on each topic to view its contents, which may include lessons, presentations, videos, quizzes, and more.



The screenshot displays the AVICO MOOC platform interface. The browser address bar shows the URL `mooc.avico-project.com/course/view.php?id=2`. The page title is "Coding training with Aviation Technologies MOOC Platform". The left sidebar contains a navigation menu with categories: "General" (Announcements, AVICO Glossary), "Course 1: Unmanned Aircraft Training" (Course 1 Description, Module 1: Technologies, Aero..., Module 2: Propulsion Systems, Module 3: RC, Avionics 03 and..., Module 4: Autonomous Flight..., Module 5: Civil Aviation Rules, Module 6: Meteorology and..., Module 7: Basic Level Design, Module 8: Maintenance and..., Module 9: Using a Drone with..., Module 10: Codable Drone...), and "Course 2: Advanced Python Programming for UAVs". The main content area is titled "Course in English" and features tabs for "Course", "Participants", "Grades", "Activities", "Competencies", and "More". A notification states "You are enrolled in the course." Below this, there are sections for "General" (Announcements, AVICO Glossary), "Course 1: Unmanned Aircraft Training" (Description, 25 Activities, Progress: 0/24), "Course 2: Advanced Python Programming for UAVs" (2 Activities, Progress: 0/2), and "Course 3: UAV Technologies, Aerodynamics and Programming".

Figure 11. Courses` content

Each course is divided into:

- modules accompanied by
- the course`s description,
- modules` objectives,
- learning outcomes,
- module organization,
- study recommendations,
- recommended time for completing the module,
- section for further thoughts and discussion.

You can access assignments and quizzes within their specific topics by clicking on the title of the assignment or quiz. This will open the activity, where you can follow the provided instructions to submit your assignment or complete the quiz.

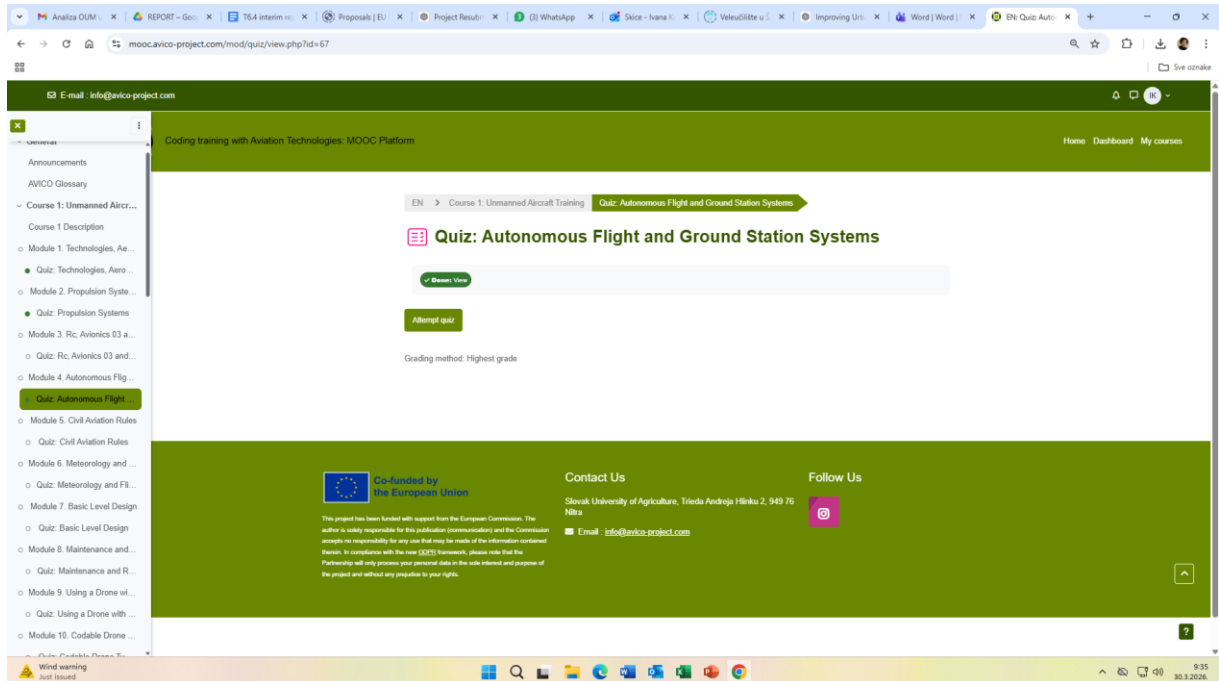


Figure 12. Quiz

2.3. Communication and Support

Stay updated by checking notifications regularly. Use them to get updates about course activities, upcoming deadlines, or important announcements. Plus, you can interact with peers and instructors by participating in discussions or asking questions directly through notification-linked forums or posts.



Figure 13. Notification section

3. Experiment Kit (Simulation) Guide

3.1. Purpose and role of the experiment Kit

The AVICO Experiment Kit is designed as a practical support component of the AVICO MOOC platform. Its purpose is to help learners apply selected theoretical knowledge from the online modules through simulation-based, task-oriented and practical learning activities.

The kit supports the connection between online learning and practical experimentation. Learners can use it to test basic UAV-related concepts, coding logic, mission planning principles, sensor-based thinking and data interpretation in a structured and safe learning environment. Depending on the available equipment and institutional conditions, the activities may be implemented through simulation tools, programmable educational drones, classroom exercises, data analysis tasks or guided project work.

Rather than functioning only as a physical set of equipment, the Experiment Kit should be understood as a methodological and practical learning resource. It provides teachers and trainers with examples of how the AVICO course content can be transferred into practical exercises suitable for VET learners.

Core objectives

- **Linking theory with practice:** The kit helps learners move from theoretical concepts presented in the MOOC to practical UAV-related tasks, such as basic flight logic, coding exercises, mission planning or data interpretation.
- **Supporting safe experimentation:** Simulation-based activities allow learners to test mission logic, identify mistakes and improve their solutions before working with real equipment, where available.
- **Connecting activities with the AVICO curriculum:** The practical tasks are linked to the main thematic areas of the AVICO syllabus, including UAV principles, coding, GIS, data use and applications of drones in agriculture and related sectors.
- **Developing relevant VET competences:** The kit supports the development of practical and technical competences at EQF 3–4 level, especially problem-solving, basic programming logic, data interpretation, technical awareness and responsible decision-making.
- **Supporting teachers and trainers:** The kit provides a flexible framework that can be adapted to different learning environments, including schools with access to drone equipment as well as institutions relying mainly on simulations or classroom-based activities.



3.2. Pedagogical use scenario

The AVICO Experiment Kit is intended to support the practical implementation of the AVICO MOOC in vocational education and training. It provides teachers and trainers with a flexible framework for combining online theoretical learning with practical, simulation-based or classroom-based activities.

The pedagogical approach is based on gradual learning. Learners first become familiar with basic concepts through the MOOC platform and then apply selected knowledge in guided practical tasks. These tasks may include coding exercises, simple mission planning, interpretation of UAV-related data, GIS activities or simulation scenarios. The level of practical implementation can be adapted according to the available equipment, digital tools and teaching conditions of each VET provider.

Target audience

The Experiment Kit is primarily designed for:

- VET students in fields such as aeronautics, ICT, electronics, agriculture, agri-food technologies or related technical disciplines;
- teachers and trainers who wish to include UAV and coding-related activities in their courses;
- novice UAV operators who need structured introductory training;
- learners and professionals from the agri-food sector who are interested in the practical use of drones and digital technologies.

Learner entry level

No advanced technical or UAV experience is required to start working with the Experiment Kit. The activities are designed to be accessible to learners with different levels of prior knowledge.

For introductory activities, learners should be able to understand basic instructions, follow a step-by-step procedure and work with simple digital tools. For coding-related activities, basic logical thinking is sufficient at the beginning. More advanced tasks, especially those linked to Python or GIS, may require basic familiarity with programming syntax, data handling or map-based tools.

Teachers may adjust the difficulty of the activities according to the learners' profile. For beginners, simulation-based exercises and block-based coding are recommended. For more advanced learners, the same tasks can be extended through Python scripting, data analysis or more complex mission planning.

Recommended duration

The Experiment Kit can be implemented flexibly, depending on the structure of the course and the time available. As a general recommendation, the full set of activities may require



approximately **80–100 hours**, including theoretical preparation through the MOOC and related practical work.

This may be divided into approximately **15–20 hours per course module**, depending on the number of selected tasks and the depth of implementation. VET providers may also use only selected activities from the kit as short practical exercises, project-based assignments or complementary tasks within existing subjects.

Delivery model

The recommended delivery model is **blended learning**. The theoretical part is delivered through the AVICO Moodle MOOC platform, while the Experiment Kit supports practical application of the acquired knowledge.

The practical part may be organised in different formats:

- classroom-based activities using worksheets, datasets or guided exercises;
- simulation-based activities using available software tools;
- laboratory sessions with programmable educational drones or other available equipment;
- project-based group work focused on solving a defined UAV-related task;
- individual coding, GIS or data interpretation exercises.

Group work and individual activities

The Experiment Kit combines individual and collaborative learning. Activities that involve planning, discussion, safety checks, mission design or hardware-related work are suitable for small groups of **2–3 learners**. This supports teamwork, communication and role division.

Coding, GIS and data analysis activities may be carried out individually, especially when the aim is to assess each learner's understanding and technical progress. However, peer discussion and comparison of results are encouraged, as they help learners reflect on different approaches and improve their solutions.

This structure allows the Experiment Kit to be used in different educational settings, from short classroom demonstrations to more complex practical training sessions integrated into VET curricula.

3.3. Technical specifications

This section defines the essential technological infrastructure required to deploy and operate the AVICO Experiment Kit effectively. It outlines the software environments, hardware components, and computing benchmarks that constitute the operational backbone of the course. These specifications are designed to ensure that students and instructors have the necessary tools for simulation, coding, and spatial data analysis, while maintaining high



standards of hardware compatibility and system reliability across all experimental tasks.

3.3.1. Software requirements

The digital tools listed below represent the integrated software suite necessary for the development, simulation, and analytical phases of the AVICO course. These applications range from visual block-coding environments for beginners to professional-grade Python distributions and Geographic Information Systems (GIS). Proper configuration of these requirements is essential for establishing a seamless workflow between the virtual laboratory and the physical aircraft.

- **Simulation platforms:**

- **DJI Tello EDU Simulator:** A lightweight, entry-level environment specifically designed for educational purposes. It is the primary tool for testing block-based logic (Course 1) and basic Python commands. It provides immediate visual feedback on flight paths, allowing students to visualize "The Perfect Square" (ST-01) or basic waypoint navigation without the risk of hardware damage.
- **Gazebo:** A professional-grade, high-fidelity physics simulator often paired with the Robot Operating System (ROS). It allows for the simulation of complex rigid-body dynamics, which is crucial for Course 3 (Flight Mechanics). Students can use Gazebo to test how sensor noise affects the "Sensory Fusion Hub" (ST-11) or to model the impact of air density on lift.
- **AirSim:** Developed by Microsoft and based on the Unreal Engine, AirSim provides photorealistic rendering and advanced AI capabilities. It is the preferred environment for Course 1, Module 12 (AI Tracking) and Course 5 (Agricultural applications). Its high visual fidelity allows for the training of computer vision models used in the "AI Object Tracker" (ST-08) and "Cattle Guardian" (ST-12) scenarios.

- **Coding environments:**

The coding stack is designed to scale with the learner's proficiency, moving from visual logic to professional scripting:

- **Python 3.x (Anaconda Distribution):** Chosen for its dominance in the UAV and Data Science industries. The Anaconda distribution is recommended as it pre-bundles essential libraries like NumPy, Pandas, and Matplotlib, simplifying environment management. This allows students to focus immediately on Course 2 (Python for UAVs) and Course 3 (Programming) without resolving complex dependency conflicts.
- **Jupyter Notebooks:** Serves as the primary platform for data analysis and experimentation. Its "literate programming" approach allows students to combine live code with equations (LaTeX) and visualizations in a single document. This is instrumental for recording mission hypotheses (Phase A) and



generating visual outputs during Data Processing (Phase D).

- **Scratch & DroneBlocks:** Utilized in the introductory modules (Course 1, Mod 11). These block-based languages allow students to master fundamental computational concepts—such as loops, conditionals, and variables—through a drag-and-drop interface. This ensures that students can execute mission logic like "The Perfect Square" (ST-01) regardless of their initial coding background.

- **GIS and mapping:**

QGIS (Open Source) for spatial analysis: As the industry-standard open-source Geographic Information System, QGIS is the core analytical engine for Course 4 and Course 5. Its inclusion in the kit allows students to process raw aerial data into actionable intelligence without licensing barriers.

- **Photogrammetric Output Management:** QGIS is used to import and visualize Orthomosaics (high-resolution maps made from stitched drone photos). Students learn to georeference these maps to ensure they align perfectly with real-world coordinates.
- **Terrain and Elevation Modeling:** It enables the generation of Digital Elevation Models (DEM) and Digital Surface Models (DSM). These are vital for "Digital Cartographer" (ST-03) and for calculating volume or slope in "Forest Restoration" (ST-09) tasks.
- **Precision Agriculture Tools:** Using the Raster Calculator, students implement spectral index formulas like NDVI (Normalized Difference Vegetation Index). This is the key to "Crop Stress Hunter" (ST-04), allowing the identification of plant health issues from multispectral simulated data.
- **Spatial Interoperability:** QGIS supports the integration of satellite imagery, vector data (cadastral boundaries), and real-time drone telemetry, providing a holistic view for tasks like "Cattle Guardian" (ST-12).

- **Ground Control Station (GCS) Software:**

Acting as the central command-and-control interface, GCS software establishes bidirectional communication between the operator and the UAV via the MAVLink protocol. This is the primary tool for Course 1, Module 4 (Autonomous Flight and Ground Station Systems).

- **Mission Planner:** An advanced, Windows-based ground station for the ArduPilot ecosystem. It provides the most comprehensive set of tools for deep hardware configuration, PID tuning, 3D waypoint planning, and forensic flight log analysis. It is essential for tasks requiring granular adjustments to flight mechanics (ST-11).
- **QGroundControl:** A modern, cross-platform GCS designed for high usability and touch-optimized interfaces. It offers a standardized experience for both PX4 and ArduPilot systems. In the kit, QGroundControl is used for designing automated grid surveys and monitoring live sensor telemetry during agricultural simulation



scenarios (ST-04, ST-09, ST-12).

3.3.2. Hardware components

The physical components of the Experiment Kit are designed to provide a tangible platform for testing the theoretical concepts and code developed during the course. This hardware selection transitions the student from the virtual safety of a simulator to the complexities of real-world physics, aerodynamics, and sensor data acquisition. By assembling, calibrating, and deploying these specific modules, learners gain essential hands-on experience in UAV maintenance, flight mechanics, and precision electronic systems.

- **UAV systems:**

The kit includes two distinct aerial platforms to cover the full spectrum of the AviCo curriculum:

- **Programmable Mini-Drone (Tello EDU Style):** This platform is dedicated to the software development modules (Course 1, Module 11; Course 2). It features an open SDK that allows students to execute commands directly via Python or block-coding. Its internal stabilization and indoor safety features make it the ideal tool for practicing mission logic, autonomous navigation, and basic computer vision tasks without the need for extensive flight regulations.
- **DIY Assembly Quadcopter Kit (5-inch Frame):** A professional-grade assembly kit focused on the hardware and engineering modules (Course 1, Modules 1, 2, 8; Course 3). This "Barebone" kit requires students to physically build the drone, providing direct experience with Electronic Speed Controllers (ESCs), Brushless Motors, Flight Controllers (FC), and Power Distribution Boards (PDB). It is the primary tool for learning maintenance, repair, and the underlying electronics of professional UAV systems.

- **Advanced Sensor Suite:**

The kit features a high-precision sensor array that enables autonomous navigation and specialized data collection. These sensors are the "eyes and ears" of the drone, and mastering their integration is a core learning outcome:

- **GPS Module (Global Positioning System):** Essential for outdoor waypoint navigation and georeferencing aerial imagery. It allows the drone to hold its position in the wind and enables the "Return to Home" (RTH) failsafe. In Course 4, GPS data is the foundation for GIS mapping and digital cartography.
- **IMU (Inertial Measurement Unit):** Comprising a 3-axis Accelerometer and a 3-axis Gyroscope. This is the heart of flight stabilization. The IMU measures tilt, acceleration, and rotation rates at hundreds of cycles per second. Students analyze IMU logs in Course 3 to understand flight dynamics and vibration dampening.



- **Barometer (Atmospheric Pressure Sensor):** Used for precision altitude-hold. By detecting minute changes in air pressure, the barometer allows the drone to maintain a consistent height relative to takeoff. This is critical for orthomosaic image capture (Course 4) where consistent altitude is required for accurate stitching.
- **NoIR Camera (Near-Infrared):** A specialized camera without an infrared filter, enabling the capture of NIR wavelengths. When combined with visible light data, it allows for the calculation of the NDVI index. This sensor is the primary tool for Course 5, Module 2, enabling students to simulate and perform precision agriculture monitoring.
- **Workstation computing requirements:** The computational power required by the kit is driven by the need for low-latency simulation physics and the heavy processing of spatial data:
 - **Processor (Minimum Intel i5 or equivalent):** A multi-core processor is mandatory to handle concurrent threads from the Ground Control Station, the flight simulator (Gazebo/AirSim), and the Python script. An i5 processor (10th Gen or newer) ensures that the high-frequency telemetry data does not cause system lag, which could lead to critical "fly-away" scenarios during testing.
 - **Memory (8GB RAM Minimum, 16GB recommended):** 8GB is the baseline for running basic simulators and coding IDEs. However, 16GB is strongly recommended for Course 4 (GIS) and Course 5 (Agriculture). Photogrammetry tasks—such as stitching hundreds of aerial photos into a high-resolution Orthomosaic—are extremely memory-intensive.
 - **Operating System (Windows 10/11 or Linux Ubuntu 20.04+):** * **Windows:** Required for professional GCS tools like Mission Planner and the DJI Tello EDU SDK, offering the best driver support for most commercial hardware.
 - **Linux (Ubuntu):** The industry standard for advanced UAV research. Ubuntu is essential for students moving into Course 3, as it provides native support for ROS (Robot Operating System) and Gazebo, allowing for more complex autonomous logic development.
 - **Graphics (Dedicated GPU recommended):** While integrated graphics may suffice for simple tasks, a dedicated GPU (e.g., NVIDIA GTX/RTX series) significantly accelerates the rendering of 3D maps in QGIS and improves the physics fidelity in AirSim.



3.4. Structure of experimental tasks

Every task within the kit follows a standardized 5-phase protocol designed to mirror professional engineering workflows:

1. Phase A: Concept – Problem framing and theoretical foundation

- **Problem identification:** Students move from a general query (e.g., "How do we stabilize flight?") to a specific technical challenge. This involves identifying the variables at play—such as wind resistance, sensor noise, or battery lag.
- **Theoretical linkage:** Students must map the problem to the specific AviCo modules. For example, stabilization relates to *Course 1: Aerodynamics* and *Course 3: Flight Mechanics*.
- **Hypothesis development:** Based on the curriculum, students predict the outcome. ("If we increase the Derivative gain in the PID controller, the drone will react faster to sudden gusts").
- **Defining constraints:** Identifying the limits of the equipment (e.g., sensor refresh rates or flight time limits) to ensure the proposed solution is realistic.

2. Phase B: Setup – Safety and environment configuration

- **Hardware Pre-flight checklist:** A physical audit of the aircraft. This includes verifying structural integrity (screws and frame), checking motor rotation smoothness, and ensuring propellers are mounted in the correct CW/CCW orientation. Battery voltage must be verified using a multimeter or telemetry.
- **Software environment initialization:** Preparing the digital workspace. This involves opening the necessary IDE (e.g., Jupyter Notebook), importing libraries (NumPy, Matplotlib), and establishing a stable serial or Wi-Fi link between the workstation and the UAV/Simulator.
- **Sensor calibration:** Executing calibration routines for the IMU (leveling) and the Compass (away from magnetic interference). For GIS tasks, this includes setting the correct Coordinate Reference System (CRS) in QGIS.
- **Operational safety zone:** Defining the "Flight Volume." For indoor tests, setting up safety nets or boundaries; for outdoor tests, verifying GPS satellite count (>6) and checking for local "No-Fly Zone" restrictions.
- **Failsafe verification:** Confirming that "Return to Home" (RTH) or "Auto-Land" triggers are correctly coded and active in case of signal loss or critical low battery.

3. Phase C: Action – Implementation and mission execution

- **Iterative coding workflow:** Developing the script in small, testable blocks. Students implement the logic (e.g., a flight path or a data processing loop) and deploy it to the simulator first. If successful, the code is uploaded to the physical UAV.
- **Mission initiation:** Executing the "Arming" sequence and starting the flight. For



autonomous tasks, this is the activation of the script; for manual tasks, this is the start of the piloting exercise.

- **Real-time telemetry monitoring:** Actively observing the "Live Feed" of flight data on the Ground Station. Students must track pitch, roll, yaw, altitude, and current draw. This is critical for detecting deviations from the Phase A hypothesis in real-time.
- **Dynamic adjustment:** Identifying and correcting errors "on the fly." If the drone behaves unexpectedly, students must decide whether to continue the mission with manual overrides or trigger an emergency land.
- **Data capture:** Ensuring that all sensors (Camera, GPS, IMU) are correctly logging data to the internal SD card or cloud-linked telemetry server throughout the duration of the "Action."

4. Phase D: Data processing – Analysis and visualization

- **Data retrieval and parsing:** Extracting raw log files (CSV, BIN, or TLOG) from the flight controller's SD card or Ground Station software. Images captured during the mission are offloaded for photogrammetric processing.
- **Data cleaning and preparation:** Using Python (Pandas library) to filter noise, handle missing data points (e.g., GPS signal drops), and synchronize timestamps between different sensors (IMU vs. GPS).
- **Advanced analysis (feature engineering):** Transforming raw metrics into technical indicators. This includes calculating the Lift Coefficient (C_L), finding the "Drag Polar" (Course 2, Mod 8), or computing the Normalized Difference Vegetation Index (NDVI) from multispectral imagery for Course 5 tasks.
- **Professional visualization:** Generating informative plots and maps. Students use Matplotlib to create time-series graphs of motor efficiency or QGIS to process raw photos into 3D Point Clouds, Digital Elevation Models (DEM), and Orthomosaics.
- **Verification of results:** Comparing the processed data against the Phase A hypothesis. Students determine if the observed physical behavior correlates with the expected mathematical model.

5. Phase E: Conclusion – Reflective debriefing and optimization

- **Technical comparison:** A structured comparison between the "Expected Results" (Hypothesis from Phase A) and the "Actual Observed Data" (Results from Phase D). Students must quantify errors or discrepancies.
- **Root Cause Analysis (RCA):** If the drone's behavior deviated from the model (e.g., unstable flight, inaccurate GPS waypoints), students investigate why. Common causes include environmental noise, sensor drift, or latency in the Python execution loop.
- **Peer review and presentation:** Sharing findings with the class. Students explain

their methodology and defend their data interpretations, simulating a professional engineering debriefing session.

- **System optimization proposals:** Based on the analysis, students propose specific code or hardware changes to improve performance in future missions (e.g., "Adjusting the PID Integral gain to reduce steady-state error").
- **Final lab documentation:** Compiling the work into a standardized report (see Annex) that includes the finalized code, key visualizations, and a signed checklist confirming the safe storage of the Experiment Kit.

3.5. Practical simulation scenarios

This section presents a curated collection of practical simulation tasks and hands-on experiments. These scenarios are designed to bridge the gap between theoretical modules and professional field applications. By engaging with these diverse tasks—ranging from basic flight mechanics to advanced AI-driven monitoring—students develop the critical thinking and technical proficiency required in the evolving UAV industry. Each scenario is mapped to specific curriculum modules to ensure a coherent learning journey that translates coding logic into physical movement and environmental data.

Linked to the AVICO Curriculum modules Table 1:

Task ID	Title	Course Link	Objective	Expected Output
ST-01	The Perfect Square	Course 1 / Mod 11	Master block-based logic for trajectory control.	Automated takeoff, square path, and precision landing.
ST-02	Aerodynamic Polar Tracker	Course 2 / Mod 8	Use Python (Matplotlib) to track and plot the aerodynamic polar (Cl vs Cd).	A drag polar plot illustrating aerodynamic efficiency and stall points.
ST-03	Digital Cartographer	Course 4 / Mod 1	Create a 3D terrain model from simulated aerial images.	A DEM (Digital Elevation Model) file in QGIS.
ST-04	Crop Stress	Course 5 / Mod 2	Identify "yellow	A prescription

	Hunter		zones" in a field using simulated NDVI data.	map for targeted irrigation.
ST-05	Emergency Fail-Safe	Course 6 / Mod 1	Code a "Return to Home" (RTH) trigger for low battery.	Drone automatically returns when battery < 20%.
ST-06	Lift & Drag Calculator	Course 1 / Mod 1	Apply Bernoulli's equation to model flight performance.	A Python script calculating real-time lift based on wing area.
ST-07	Maintenance OOP	Course 2 / Mod 3	Use Object-Oriented Programming to track hardware health.	A class-based system notifying the user of motor wear.
ST-08	AI Object Tracker	Course 1 / Mod 12	Implement computer vision (OpenCV) for target following.	Simulation video showing the drone tracking a moving car.
ST-09	Forest Restoration	Course 5 / Mod 2	Plan a seed-bombing mission for post-fire recovery areas.	A flight plan optimized for terrain elevation and shadows.
ST-10	Regulatory Auditor	Course 6 / Mod 1	Create a digital audit for Slovakia/EU drone compliance.	A software checklist cross-referenced with the Aviation Act.
ST-11	Sensory Fusion Hub	Course 1 / Mod 4	Combine GPS, IMU, and Barometer data for positioning.	A fused dataset showing high-accuracy altitude estimation.
ST-12	Cattle Guardian	Course 5 / Mod 2	Simulate animal monitoring using thermal imaging	A heat-map visualization identifying

3.6. Assessment and evaluation

Performance is measured through a multi-dimensional rubric:

- **Safety (30%):** Adherence to pre-flight checklists and battery handling protocols.
- **Technical accuracy (30%):** Does the code run without errors? Is the drone assembly structurally sound?
- **Analytical depth (20%):** Ability to interpret GIS maps or Python data logs correctly.
- **Documentation (20%):** Quality of the mission report and flight log entries.

3.6.1. Scenario-specific assessment rubrics

Table 2. For each scenario, the following criteria apply based on the weighted percentages above:

Task ID	Safety (30%)	Technical Accuracy (30%)	Analytical Depth (20%)	Documentation (20%)
ST-01	Correct takeoff/landing protocols; no collisions.	Logic of the square path loop is correct; precise landing.	Identification of drift causes (wind/sensors).	Complete block-code screenshot and flight log.
ST-02	Secure data export; proper handling of simulator files.	Correct Matplotlib plot syntax; accurate Cl/Cd calculation.	Interpretation of the "Stall Point" and "Best Glide" ratio.	Comparative analysis of different airfoil types.
ST-03	Mission area safety checks; safe altitude planning.	Orthomosaic overlap quality (>70%); QGIS setup accuracy.	Accuracy of terrain height measurements.	Exported DEM report and topographical map.
ST-04	Ground-truthing safety; sensor calibration checks.	Correct NDVI formula implementation; spectral alignment.	Correlation between NDVI values and plant health.	Prescription map with irrigation recommendations.
ST-05	Effectiveness of the RTH trigger;	Code logic for low-battery	Justification of the 20% safety	Video of successful

	obstacle avoidance.	detection; GPS return precision.	threshold.	autonomous return and log data.
ST-06	Validation of mathematical limits; unit consistency.	Correct application of Bernoulli's formula in code.	Sensitivity analysis of lift vs. air density.	Table of results for varying flight conditions.
ST-10	Accuracy of regulatory data; GDPR compliance check.	Completeness of the auditor tool; database logic.	Legal interpretation of "No-Fly Zones" and permissions.	Final compliance report for a specific flight scenario.
ST-11	IMU/GPS signal monitoring; sensor health checks.	Fusion algorithm logic (Kalman/Complementary); drift fix.	Comparison between raw data vs. fused data quality.	Log analysis showing reduced altitude error.
ST-12	Animal safety distance; thermal sensor calibration.	Thermal detection algorithm logic; heat-map generation.	Accurate identification of "hot spots" vs. environment.	Visualization of herd distribution and welfare report.

3.7. Teacher's manual and facilitation

This section provides essential guidance for educators and trainers responsible for delivering the AVICO curriculum. It outlines pedagogical strategies designed to maximize student engagement and critical thinking while managing the technical complexities of UAV flight and coding. The manual serves as a roadmap for facilitating hands-on sessions, ensuring that instructors can transition from being mere sources of information to active guides in the students' problem-solving journey.

3.7.1. Facilitation strategies

Facilitation strategies focus on shifting the instructor's role from a primary information provider to a mentor and catalyst for discovery. By employing the Socratic method, teachers challenge students to articulate the physics and logic behind their UAV's behavior rather than providing immediate technical fixes. Furthermore, role rotation ensures that every student



experiences the diverse responsibilities of a professional drone team—from the precision of the pilot to the analytical rigor of the data scientist—fostering a collaborative environment where technical skills and soft skills are developed simultaneously.

- **The Socratic Method:** When a student's drone drifts, ask: *"Why do you think the drone drifts left?"* instead of giving the answer.
- **Role rotation:** Ensure students swap between *Pilot*, *Safety Observer*, and *Data Analyst* roles.
- **"Fail-Fast" encouragement:** Create a culture where students are encouraged to fail in the simulator first. Use crashes as debugging sessions to understand the limits of their code.
- **Peer-to-Peer code review:** Before physical flight, have students exchange their Python scripts. If one student can't explain the logic to another, the script isn't ready for deployment.
- **Sim-to-Real validation:** Mandate that no code can be uploaded to a drone until it has successfully completed the mission three times in a row within the flight simulator.
- **Safety ethics advocacy:** Constantly bridge technical tasks with legal/ethical implications (e.g., *"If your Cattle Guardian AI misidentifies a target, what is the legal liability based on Course 6?"*).

3.7.2. Troubleshooting and common errors

This section serves as a practical diagnostic guide to help users navigate the technical challenges that may arise during hardware assembly, software development, or field operations. By identifying and categorizing the most frequent points of failure, this resource aims to streamline the troubleshooting process, allowing students to learn from common mistakes and maintain high operational readiness.

- **Hardware:**
 - **Inverted propellers:** The #1 cause of failed takeoffs. Verify CW/CCW orientation markings.
 - **GPS lock failures:** Ensure you are outdoors with a clear view of the sky; avoid testing under heavy tree cover or near tall skyscrapers.
 - **Compass interference:** Do not calibrate the drone near large metal objects (cars, reinforced concrete) or high-voltage lines.
 - **LiPo voltage sag:** If the drone behaves erratically during high throttle, check for battery wear or low storage voltage.
- **Software:**
 - **Python indentation:** Critical in Course 2. Use a linter or IDE that visually highlights tabs vs. spaces.
 - **Port/IP conflicts:** If the script cannot find the UAV, ensure no other background



- processes are using the same communication port.
- **Missing dependencies:** "ModuleNotFoundError" usually means NumPy, Pandas, or OpenCV need to be installed via pip.
- **Infinite SDK loops:** Always include "break" or "timeout" conditions in autonomous scripts to prevent "fly-away" scenarios.
- **GIS:**
 - **CRS mismatches:** Mismatched Coordinate Reference Systems cause misaligned map layers. Always verify EPSG codes in QGIS project settings.
 - **Photogrammetry holes:** Usually caused by insufficient overlap (<70%). Re-run the mission with a tighter flight path grid.
 - **Z-Data missing:** Ensure the raster layer contains elevation values before attempting 3D terrain rendering.

3.8. Annex: Ready-to-use materials

The Annex provides a collection of standardized forms, templates, and observation sheets designed to support the pedagogical and operational requirements of the AVICO course. These materials act as a "ready-to-use" toolkit for teachers and students, ensuring consistency in data collection, safety management, and performance evaluation. By utilizing these standardized resources, vocational schools can maintain high professional standards while streamlining the administrative aspects of laboratory and field activities.

3.8.1. Checklist template: Pre-flight safety

- ✓ Visual inspection of frame and propellers.
- ✓ Battery voltage checked (>3.8V per cell).
- ✓ Radio signal strength confirmed.
- ✓ No-fly zone status verified (Legislation Module).
- ✓ Compass calibration successful.

3.8.2. Observation sheet: data analysis

- **Student Name:** _____
- **Task:** ST-02 (Aerodynamic Polar Tracker)
- **Observed Behavior:** _____
- **Result:** [Pass / Fail] _____

4. Adoption and replication framework

4.1 Purpose and strategic approach

The purpose of this framework is to support Vocational Education and Training (VET) providers in effectively adopting, adapting, and implementing the AVICO learning approach within their own institutional contexts. Rather than presenting a rigid implementation model, the AVICO framework is designed as a flexible and transferable methodology that can respond to the diverse needs of learners, teachers, and educational systems across Europe.

In the context of rapidly evolving labour market demands and digital transformation, VET providers are increasingly required to integrate innovative teaching approaches that combine technical knowledge with practical application. The AVICO framework addresses this challenge by offering a structured yet adaptable model that connects coding education with real-world UAV applications, enabling institutions to modernise their training offer while maintaining alignment with their existing curricula.

The framework therefore serves a dual function: it provides practical guidance for implementation, while also supporting strategic decision-making at institutional level, particularly in relation to curriculum development, digitalisation, and skills innovation.

4.2. Adoption process and integration into VET practice

The adoption of the AVICO approach is conceived as a progressive and manageable process that allows VET providers to integrate the MOOC platform and simulation-based activities into their teaching practices without requiring major structural changes.

Institutions are encouraged to begin by exploring the AVICO MOOC platform and familiarising themselves with the available modules, learning outcomes, and resources. Based on this initial exploration, teachers and trainers can identify the most relevant components in relation to their learners' needs, the level of digital competence, and the specific objectives of their training programmes.

The integration process typically involves combining theoretical learning delivered through the MOOC with practical activities supported by the simulation-based experiment kit. This blended approach enables learners to move from knowledge acquisition to application, reinforcing understanding and supporting the development of technical and transversal skills.

Importantly, the AVICO model allows for gradual implementation. VET providers can start with selected modules or pilot activities and progressively expand their use of the tools as confidence and experience grow. This incremental approach reduces barriers to adoption and supports sustainable integration into everyday teaching practice.



4.3. Adaptation to learners' needs and implementation models

A key strength of the AVICO framework lies in its adaptability. The methodology is intentionally designed to accommodate a wide range of learner profiles, institutional settings, and educational objectives, making it particularly suitable for the diverse landscape of VET systems.

VET providers can tailor the AVICO approach by selecting specific modules, adjusting the level of complexity, and contextualising activities to reflect local needs or sector-specific requirements. This ensures that the learning experience remains relevant, inclusive, and aligned with the competences that learners are expected to develop.

The framework supports multiple implementation models, including classroom-based learning, blended learning environments, and project-based or extracurricular activities. This flexibility allows institutions to choose the most appropriate format depending on available resources, infrastructure, and pedagogical preferences.

By focusing on a learner-centred approach and offering adaptable pathways, AVICO enables VET providers to effectively address different levels of digital readiness and learning styles, while maintaining a coherent and structured educational experience.

5. Sustainability and long-term use

The AVICO framework has been developed with sustainability as a core principle, ensuring that the project results remain relevant and usable beyond the formal duration of the project. The open and modular nature of the MOOC platform and simulation tools allows VET providers to continue using, adapting, and integrating the resources into their educational programmes over time.

Sustainability is further supported by the emphasis on capacity building among teachers and trainers, who play a key role in embedding the AVICO methodology within their institutions. By developing familiarity with the tools and approaches, educators can continue to apply and evolve the methodology in response to changing educational needs and technological developments. In addition, the dissemination strategy contributes to long-term impact by promoting awareness, encouraging adoption, and fostering a community of practice among VET providers and stakeholders. This network-based approach enhances knowledge sharing, supports continuous improvement, and strengthens the overall scalability of the AVICO results at national and European level.

Through this combination of flexibility, accessibility, and institutional integration, the AVICO framework ensures that its impact extends beyond the project lifecycle, contributing to the ongoing modernisation of vocational education and training.