



Deliverable D5.1 Model Repository UI

Deliverable type	OTHER
Dissemination level	PU - Public
Due date (month)	M25
Delivery submission date	28 February 2025
Work package number	WP5
Lead beneficiary	IOTIQ GmbH (IOTIQ)



This project has received funding from the Horizon Europe Framework Programme of the European Union under grant agreement No. 101094428

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Document Information

Project number	101094428	Acronym	CULTURATI
Project name	Customized Games and Routes For Cultural Heritage and Arts		
Call	HORIZON-CL2-2022-HERITAGE-01		
Topic	HORIZON-CL2-2022-HERITAGE-01-02		
Type of action	HORIZON-RIA		
Project starting date	1 February 2023	Project duration	36 months
Project URL	http://www.culturati.eu		
Document URL	https://culturati.eu/deliverables/ https://aperta.ulakbim.gov.tr/record/274339		

Deliverable number	5.1			
Deliverable name	Model Repository UI			
Work package number	WP5			
Work package name	AI Integration			
Date of delivery	Contractual	28.02.2025	Actual	28.02.2025
Version	1.0			
Lead beneficiary	IOTIQ GmbH (IOTIQ)			
Responsible author(s)	Ali Bugdayci, IOTIQ, ali@iotiq.de Metin Tekkalmaz, IOTIQ, metin@iotiq.de			
Reviewer(s)	Neşe Şahin Özçelik, Bilkent Universitesi Vakif, nozcelik@bilkent.edu.tr Eda Gürel, Bilkent Universitesi Vakif, eda@tourism.bilkent.edu.tr			

Short Description	This document details the core features of Model Repository UI for CULTURATI platform. It outlines key aspects and improvements in the configuration of AI models and their settings. It serves as an update on the platform's functionality and enhancements.
-------------------	--

History of Changes			
Date	Version	Author	Remarks
27 February 2025	0.1	Ali Bugdayci	First version
6 March 2025	1.0	Eda Gürel	Formatted

Executive Summary

This document serves as a technical blueprint for the configuration of our system. The platform is engineered to deliver intelligent path recommendations, generate questions, and perform high-quality transcription. It details the customizable parameters across three core modules:

- **Path Recommendation Weights:** The system dynamically calculates optimal routes by balancing congestion, proximity, and content similarity. By assigning adjustable weights to these factors, the module ensures that recommendations are tailored to avoid overcrowded areas, favors nearby locations, and aligns with the user's past interactions.
- **Question Generation Module:** This component governs the production of questions from input text. It supports a limit on the maximum number of questions generated and offers a choice between two advanced models.
- **Transcription Configuration:** Detailed settings in this section define the model's performance for audio transcription. It includes options for compute precision (ranging from INT8 to F32) and a suite of voice activity detection filters, such as beam size, silence/speech durations, and threshold levels. These settings optimize performance according to the deployment environment, whether on CPU or GPU.

By outlining these configuration options, this document empowers developers and system integrators to optimize the platform for efficiency, accuracy, and scalability, ultimately enhancing user experience across diverse applications.

Table of Contents

Executive Summary.....	3
Table of Contents.....	4
1. Introduction	5
2. Path Recommendation Weights	6
2.1. Congestion Weight	6
2.2 Location Weight.....	6
2.3 Similarity Weight	7
3. Question Generation.....	8
3.1 Maximum Question Count	8
3.2 Question Generation Model.....	8
4. Transcription Configuration	9
4.1 Compute Type	9
4.2 Transcription Model Variant.....	10
4.3 Voice Activity Detection (VAD) Filters	10
4.3.1 Beam Size	10
4.3.2 Minimum Silence Duration	10
4.3.3 Minimum Speech Duration	11
4.3.4 Threshold.....	11
Conclusion.....	12

1. Introduction

This document provides a comprehensive overview of the system configuration settings for our system. Designed to enhance user experience through optimized navigation, dynamic content generation, and advanced audio recognition, this platform integrates sophisticated modules for path recommendation, automated question generation, and high-performance transcription.

The configuration settings detailed herein have been carefully engineered to achieve optimal performance and efficiency. The Path Recommendation module leverages sensor data and user context to calculate dynamic weights—congestion, location, and similarity—that collectively ensure recommendations are both contextually relevant and user-centric.

Concurrently, the Question Generation module offers flexible control over output by setting limits on the number of questions generated and providing a choice between two state-of-the-art models, Leaf and QuestGen.

The **Transcription Configuration** section outlines critical parameters, including compute types and voice activity detection (VAD) filters, which affect the model's precision and segmentation of audio inputs. These settings are pivotal for balancing computational efficiency with transcription accuracy, particularly across varying deployment environments such as CPUs and GPUs.

Serving as a technical guide for developers and system integrators, this document explains the rationale behind each configuration parameter, provides detailed usage guidance and outlines best practices for maintaining a robust and scalable system. Through rigorous testing and iterative optimization, these configurations are intended to support a reliable, efficient, and user-friendly platform that meets the demands of modern multi-modal applications.

Additionally, as part of our commitment to open science practices, we are making the software available through open access. The software can be accessed at the following link:

<https://aperta.ulakbim.gov.tr/record/274339>

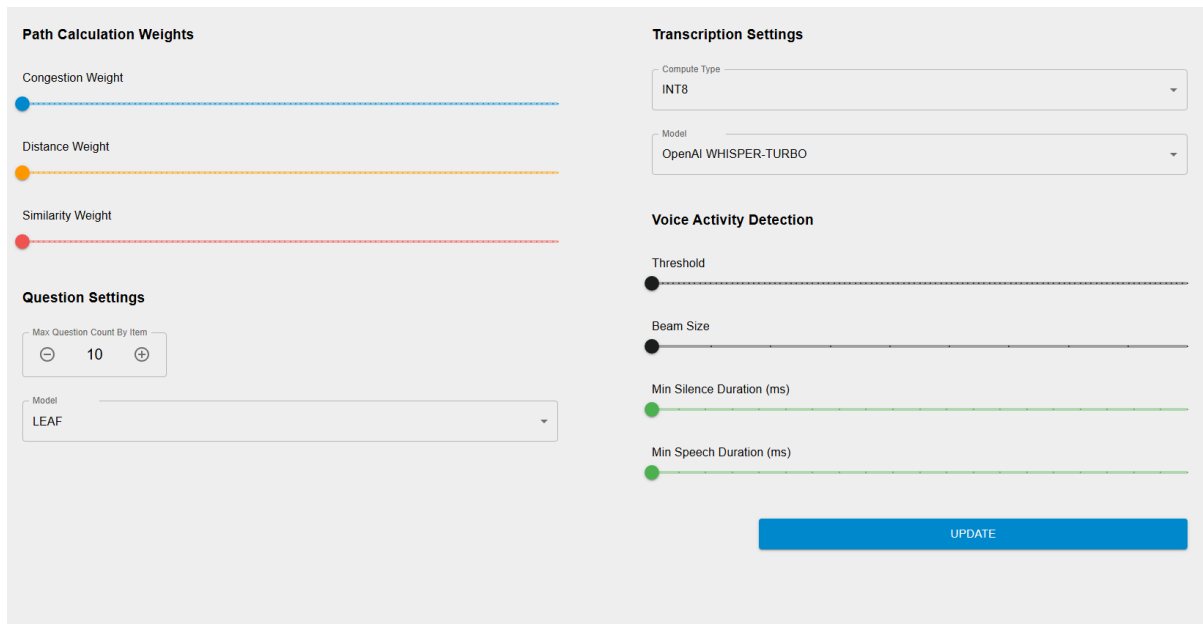


Figure 1- AI Settings

2. Path Recommendation Weights

The system leverages sensor data and real-time user context to compute optimal routes and make item suggestions. Three key weights are applied during the path calculation process to balance various factors influencing the recommendation.

2.1. Congestion Weight

- **Purpose:** Quantifies the crowd density at a given location by analyzing sensor data.
- **Effect:** A higher congestion weight drives the system to avoid crowded or busy areas, favoring routes that lead to less populated or “empty” spaces. This ensures that users experience less congestion and a more comfortable journey.

2.2 Location Weight

- **Purpose:** Measures the spatial relationship between the user’s current position and the candidate destination.
- **Effect:** With a higher location weight, the system prioritizes nearby locations in its recommendations. This means that the closer a destination is to the user, the more likely it will be selected as the recommended route, optimizing for convenience and efficiency.

2.3 Similarity Weight

- **Purpose:** This parameter defines how closely related the recommended items should be to those the user has already encountered.
- **Effect:** A higher similarity value encourages the algorithm to propose items that are more similar to the ones previously viewed by the user. This tailoring ensures that the recommendations are consistent with the user's interests.

This configuration allows the platform to dynamically adjust its recommendations, ensuring that the chosen path or item not only avoids congestion and minimizes travel distance but also resonates with the user's preferences.



Figure 2- Path Recommendation Weights Settings

Currently, the weights can be modified through the Model Repository UI, and any changes made are updated in the backend. However, the system does not yet utilize these weights within the algorithms.

At this stage, we can retrieve crowd data from two sites, with a third site nearing readiness. Our ongoing efforts are focused on integrating weight adjustments with the crowd data to enhance system functionality. Soon, these weights will be properly incorporated into the recommendation algorithms to improve accuracy and effectiveness.

3. Question Generation

This module is responsible for converting textual input into a series of contextually relevant questions. It is designed to generate high-quality quiz questions that can be used for assessment or knowledge reinforcement.

3.1 Maximum Question Count

- **Parameter:** count
- **Purpose:** Sets an upper limit on the number of questions the system is allowed to generate from the provided text.
- **Behavior:** While the system may generate fewer questions depending on the input content, it will not exceed this predefined maximum, ensuring controlled output and preventing excessive or redundant question creation.

3.2 Question Generation Model

- **Parameter:** model
- **Available Options:** Leaf/QuestGen
- **Explanation:**
 - **Leaf** is a multiple-choice question generation system that utilizes a fine-tuned T5 Transformer model. It is specifically designed to generate not only questions but also high-quality distractors from factual text, making it an excellent choice for educational and testing environments.
 - **QuestGen** is an alternative model that employs AI algorithms to generate questions. It is particularly effective at creating variations of input queries, thereby expanding question banks and offering diverse questioning styles. This model is well-suited for scenarios where generating a range of similar but distinct questions is beneficial.

Together, these configurations enable the system to produce a curated set of questions that are both contextually appropriate and aligned with the user's learning objectives.

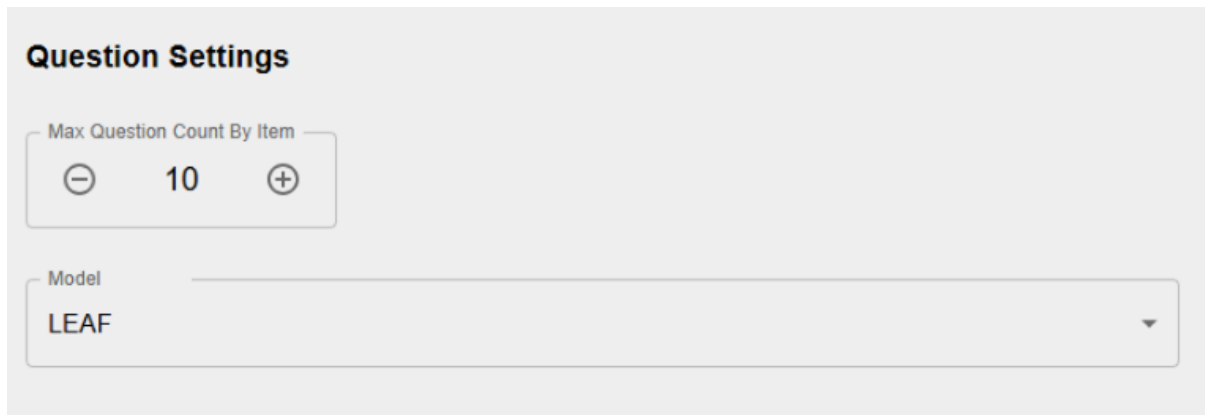


Figure 3- Question Generation Settings

4. Transcription Configuration

This module configures the settings for Transcription Service, which is employed to process and transcribe audio inputs. The configuration is split into three main areas: numerical precision (compute type), model selection, and voice activity detection (VAD) filters.

4.1 Compute Type

- **Parameter:** compute type
- **Description:** This setting defines the numerical precision format used when running the Transcription Service, which directly affects both processing speed and transcription accuracy.
- **On CPU:** Lower-precision formats such as **INT8** can accelerate inference by reducing memory usage and leveraging optimized integer operations.
- **On GPU:** Formats like **BF16 (bfloat16)** or **F16 (float16)** are often preferred to maximize performance while maintaining an acceptable balance between speed and accuracy.
- **Available Options:**
 - **INT8:** Ideal for reducing computational load on CPUs.
 - **INT16:** Provides slightly higher precision while still offering performance benefits.
 - **F16 (float16):** Enables faster computations on GPUs with half-precision.
 - **BF16 (bfloat16):** Specifically designed for deep learning on GPUs, offering a balance of dynamic range and reduced precision.
 - **F32 (float32):** Standard single-precision format, delivering the highest accuracy at the cost of increased computation.

4.2 Transcription Model Variant

- **Parameter:** model
- Available Options:
 - openai/whisper-turbo
 - openai/whisper-base
 - openai/whisper-tiny
 - openai/whisper-small
 - openai/whisper-medium
 - openai/whisper-large-v1
 - openai/whisper-large-v2
 - openai/whisper-large-v3
- **Guidance:** The **openai/whisper-turbo** variant is recommended for its balanced performance, offering a favorable trade-off between processing speed and transcription accuracy. Depending on the specific use case and available resources, other variants may be selected to optimize either speed or accuracy.

4.3 Voice Activity Detection (VAD) Filters

VAD filters help segment audio inputs into speech and non-speech segments, ensuring that only meaningful speech portions are processed.

4.3.1 Beam Size

- **Parameter:** beam_size
- **Purpose:** Specifies the number of beams or alternative hypotheses to consider during VAD filtering. Increasing the beam size may enhance detection accuracy but at the cost of higher computational overhead.

4.3.2 Minimum Silence Duration

- **Parameter:** min_silence_duration_ms
- **Purpose:** Defines the minimum duration (in milliseconds) for a silence segment to be classified as a genuine pause. This helps to clearly demarcate speech intervals and reduce false positives in speech detection.

4.3.3 Minimum Speech Duration

- **Parameter:** min_speech_duration_ms
- **Purpose:** Sets the minimum duration (in milliseconds) that a segment must sustain speech to be considered valid. This avoids misclassifying brief noises or transient sounds as speech.

4.3.4 Threshold

- **Parameter:** threshold
- **Purpose:** Establishes a confidence level that determines whether a segment contains speech. Segments scoring above the threshold are marked as speech, while those below are classified as silence, ensuring reliable segmentation.

This comprehensive configuration enables precise control over audio transcription performance, balancing efficiency and accuracy across different hardware environments and ensuring robust handling of diverse audio inputs.

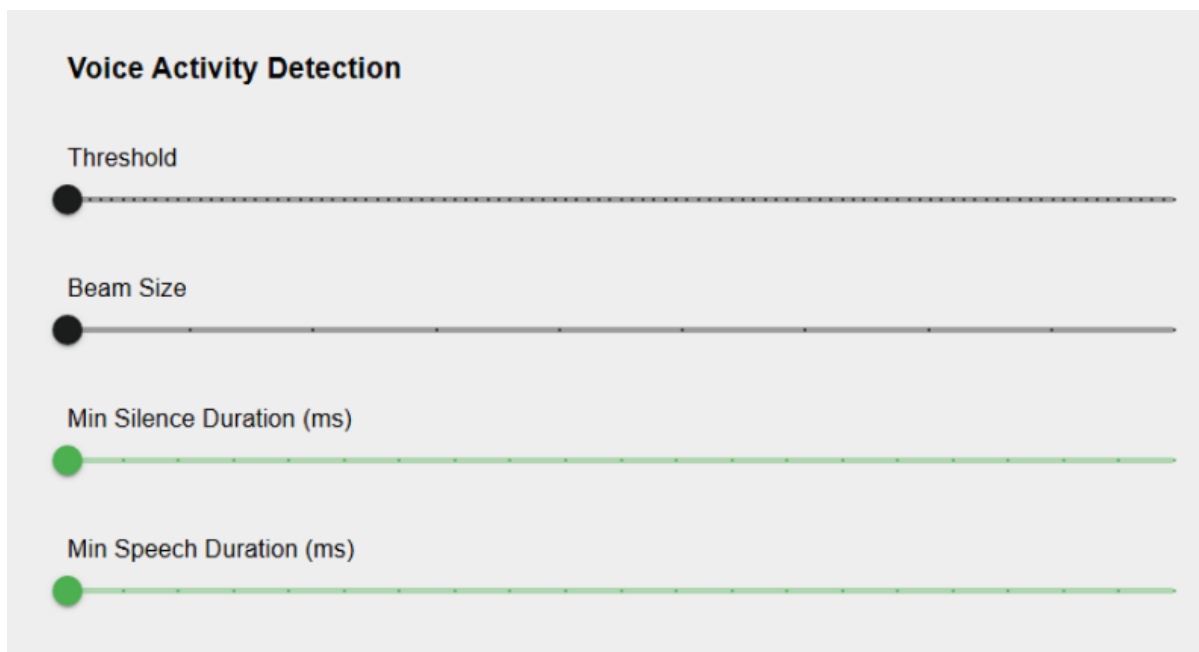


Figure 4- Speech Recognition Settings

Conclusion

This document provides a comprehensive technical framework for configuring the platform's core modules—Path Recommendation, Question Generation, and Transcription. By offering flexible and adjustable parameters, the system enables precise control over route optimization, question generation, and audio transcription, ensuring adaptability across various use cases and environments.

The integration of dynamic weighting in path recommendations helps manage congestion and personalize user experiences, while the question generation module enhances content interactivity through customizable outputs. Similarly, the transcription settings allow for performance optimization based on hardware constraints and processing needs.

By leveraging these configurable components, developers and system integrators can fine-tune the platform to meet specific operational requirements, enhancing its overall accuracy, efficiency, and scalability. As the system evolves, future iterations will further refine these features, reinforcing the platform's capability to deliver intelligent, adaptive, and high-quality services.