ISABEL

MINIMIZE COSTS AND MAXIMIZE BENEFITS IN THE TRANSITIONARY EUROPEAN REGIONAL LABOUR MARKETS

mInimize coSts and mAximize Benefits in the transitionary European regional Labour markets

Deliverable No3.5

Researcher custom-tailored tool for personalised dataset creation

WP3: Data Lake creation and Data Collection on the uneven effects of JCD

Funded by the European Union

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

DOCUMENT ID	D3.5.
DUE DATE	30/06/2026
DELIVERY DATE	30/06/2026
WORK PACKAGE	WP3
TASK(S)	T3.3
DISSEMINATION LEVEL	PU-Public
VERSION	V2
AUTHOR/S	CERTH
FILENAME	ISABEL_D3.5_v2

Abstract:
This deliverable presents the custom-tailored researcher toolkit developed under Work Package 3 of the ISABEL project. Designed to democratise complex computational methodologies, the toolkit provides a cohesive suite of interactive web environments, utilising Vue/Nuxt, Streamlit, and OpenWebUI, that enable domain experts to generate, extract, and analyse personalised labour market datasets. By abstracting the intricacies of Large Language Model (LLM) orchestration and vector mathematics behind intuitive, API-driven interfaces, the ecosystem allows researchers to actively parameterise synthetic demographic simulations, conduct citation-backed literature reviews, and extract tabular data from unstructured reports using natural language. The toolkit comprises three principal frontend components: the Persona Generator (a Vue/Nuxt Single-Page Application communicating with the SNAP Flask backend via NDJSON streaming), the Research Companion (a Streamlit application providing arXiv-based literature discovery and dual RAG-based PDF interrogation), and the Unstructured Data Extraction and Analytics Interface (hosted within OpenWebUI). All three components are described in terms of their user-facing workflows, data persistence mechanisms, and integration with the SNAP backend architecture specified in Deliverable D3.3. Crucially, the architecture enforces a strict local-first execution paradigm alongside client-side storage, guaranteeing absolute data sovereignty, privacy, and GDPR compliance. Ultimately, this deliverable demonstrates how sophisticated backend data fusion is transformed into an accessible, human-in-the-loop digital workspace, empowering researchers to investigate the European transition towards green and digital economies with autonomy and methodological transparency.

Views and opinions expressed in this document are those of the author(s) and do not necessarily reflect those of the European Union or REA. Neither the European Commission nor the granting authority can be held responsible for them.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Executive Summary

Advanced empirical research into the European labour market, particularly concerning the uneven impacts of the green and digital transitions, requires highly granular, bespoke datasets. However, researchers frequently encounter steep technical barriers when attempting to leverage modern artificial intelligence, vector databases, and programmatic data extraction. Deliverable 3.5 documents the front-end resolution to this challenge, introducing a comprehensive, custom-tailored researcher toolkit designed to facilitate the creation of personalised microdata.

Unlike Deliverable 3.3, which detailed the underlying backend orchestration and statistical taxonomies of the SNAP ecosystem, this document focuses entirely on user empowerment and interactive design. The primary objective of the toolkit is to implement a "human-in-the-loop" methodology, providing researchers with the visual interfaces necessary to inject custom hypotheses, configure simulation parameters, and interrogate raw data without writing manual Python scripts.

To accommodate diverse methodological workflows, the user environment is built upon a federated frontend architecture. This approach provides researchers with distinct, highly optimised interfaces that communicate with the unified SNAP Python orchestration backend (Flask/FastAPI, as specified in D3.3) via secure, decoupled REST APIs and NDJSON streaming. The ecosystem integrates three specialised presentation layers:

1. **The Synthetic Persona Generator (Vue/Nuxt):** A highly responsive Single-Page Application (SPA) that utilises a Nuxt Nitro server proxy to eliminate CORS issues and implements Newline-Delimited JSON (NDJSON) streaming to provide live progress monitoring. It allows researchers to upload custom survey configurations and dynamically generate statistically grounded demographic cohorts, complete with interactive visual analytics and session persistence via localStorage.

2. **The Research Companion (Streamlit):** An academic workspace dedicated to literature discovery and deep-document interrogation. It enables researchers to construct local vector libraries and execute multi-document, citation-backed Retrieval-Augmented Generation (RAG) queries.

3. **The Unstructured Data Extraction & Analytics Interface (OpenWebUI):** A conversational portal that translates natural language prompts into executable data parsing and PandasAI scripts, effectively liberating trapped tabular data from governmental PDFs and transforming them into reusable Knowledge Bases.

The architecture is consistent with the SNAP backend as described in D3.3. All frontend components communicate with the backend services documented therein: the Synthetic Persona Generator communicates with the Flask backend at persona/app.py. The Research Companion is served by research_companion/app.py, and the Extraction pipeline operates through the OpenWebUI Pipe module (exported as snap_pipe). The local-first LLM orchestration paradigm (Ollama and vLLM served locally at CERTH) described in D3.3 Section 5.1 is enforced across all three frontend components.

A defining pillar of the toolkit is its uncompromising stance on data security. Recognising the vulnerabilities of cloud-based AI processing, the entire ecosystem mandates privacy-preserving local execution. By orchestrating local LLMs and leveraging persistent client-side `localStorage`, the architecture ensures that proprietary academic literature, sensitive demographic hypotheses, and custom survey frameworks remain entirely within the secure institutional perimeter.

In conclusion, Deliverable 3.5 provides the European research community with a sophisticated, sovereign digital workspace. By enforcing execution transparency, such as surfacing the exact programmatic code used in data analysis and surfacing partial generation errors, the toolkit guarantees that researchers can curate, validate, and export personalised datasets with absolute scientific trust and reproducibility.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Contributors

NAME	ORGANISATION
Nicholas Vretos	**CERTH**
Michalis Angelou	**CERTH**
Symeon Samaras	**CERTH**
Vasileios Rekkas	**CERTH**

Peer Reviewers

NAME	ORGANISATION
Despoina Kandeler	UOWM
Luke Bosworth	IER

Versions

Version	Date	Author / Organisation	Description
ToC	15/05	CERTH	Table of Contents
V1	10/06	CERTH	Initial Version
V2	26/06	CERTH	Integrate Review comments

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Table of Contents

Executive Summary 3

1. Introduction (Scope and focus on user personalisation) *8*

2. Overview of the Researcher Toolkit *9*

2.1. Architecture of the User Environment (Front-end focus and tool synergy) 9

2.2. Privacy-Preserving Local Execution (Ensuring data security for researchers) 10

3. Personalised Synthetic Dataset Generation (Survey Simulator) *11*

3.1. Custom Survey Configuration and Upload Workflows (Uploading the survey.json via UI) 12

3.2. Live Dashboard Monitoring and Cohort Analytics (Viewing real-time generation stats) 13

3.3. Exporting Custom Microdata (Downloading the final synthetic dataset) 14

4. Interactive Literature Deep-Dive (Research Companion) *15*

4.1. Setting up Personalised Research Profiles (Entering keywords to tailor results) 16

4.2. Interacting with the RAG Pipeline (Chatting directly with specific papers) 16

4.3. Dual RAG Mode – LangChain vs. PaperQA 18

4.4. Library Management and Reset 19

5. Custom Unstructured Data Extraction *20*

5.1. User-Driven Document Selection and Prompting (Selecting PDFs and giving extraction prompts) 20

5.2. Workflow: From PDF to Personalised Knowledge Base (Getting the final structured CSV) 20

6. Conversational Data Analytics *21*

6.1. Interacting with the PandasAI Agent (Asking data questions in natural language) 21

6.2. Verifying Generated Code and Analytical Outputs (Reviewing Python code for transparency) 22

7. User Interfaces and Workspace Cohesion *23*

7.1. The Persona Generator Interface (Vue/Nuxt) 23

7.2. The Research Companion Interface (Streamlit) 23

7.3. The Extraction and Analytics Interface (OpenWebUI) 23

8. Personalised Storage and Output Management *25*

8.1. Local Vector Profiles (Managing ChromaDB user collections) 25

8.2. Managing Exported Datasets and CSVs (Organizing final outputs locally) 25

9. Validation and Output Trustworthiness *26*

9.1. Statistical Representativeness and Error Transparency in the Persona Generator 26

9.2. Citation Provenance in RAG Outputs 26

9.3. Code Logging and Analytical Reproducibility 26

10. Conclusion (Summary of researcher benefits) *27*

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Bibliography ***28***

Annexes ***29***

Annex A: Example Questionnaire JSON Schema **29**

List of Figures

Figure 1: Configuration Panel and Macro-level Cohort Analytics 11
Figure 2: Synthetic Persona Profile Grid 12
Figure 3: The Research Companion Discovery Workspace. 15
Figure 4: Deep Dive (RAG) Document Interrogation 18
Figure 5: The Extraction and Analytics Interface within OpenWebUI 22

List of Tables

Table 1. Frontend Technology Stack Summary 9
Table 2. Questionnaire Question Types and Validation Rules 13
Table 3. RAG Pipeline Comparison 18

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

List of Abbreviations

API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CSV	Comma-Separated Values
ESCO	European Skills, Competences, Qualifications and Occupations
GDPR	General Data Protection Regulation
ISCED	International Standard Classification of Education
ISCO	International Standard Classification of Occupations,
JSON	JavaScript Object Notation
KPI	Key Performance Indicator,
LLM	Large Language Model
PDF	Portable Document Format,
QA	Question Answering
RAG	Retrieval-Augmented Generation,
REST	Representational State Transfer,
SNAP	Smart oNe-stop digitAl data shoP
SPA	Single-Page Application,
UI	User Interface
vLLM	virtual Large Language Model serving framework

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

1. Introduction (Scope and focus on user personalisation)

Within the framework of the ISABEL project, the ability to generate, manipulate, and interrogate complex labour market datasets is paramount. While Deliverable 3.3 detailed the underlying backend architecture and data orchestration pipelines of the SNAP ecosystem, Deliverable 3.5 shifts the focus entirely to user empowerment. This document introduces the researcher custom-tailored toolkit, a specialised, interactive environment designed to enable researchers to curate, configure, and export highly personalised datasets without requiring deep programming expertise.

The core philosophy driving this deliverable is "human-in-the-loop" customisation. Labour market research cannot be fully automated. It requires domain experts to inject specific hypotheses, define survey constraints, and contextualise academic literature. To facilitate this, the toolkit provides a unified, intuitive user interface (UI) that abstracts the complexities of the underlying machine learning models and data fusion engines. Through this interface, researchers can upload custom survey configurations, establish specific academic retrieval profiles, and direct the extraction of tabular data from unstructured documents.
Ultimately, this deliverable enhances the overall data management framework of the ISABEL project. By bridging powerful backend algorithms with a user-friendly, API-driven front-end environment, the toolkit provides researchers with the flexibility, transparency, and precision required to generate custom-tailored microdata and derive actionable insights regarding the European transition towards green and digital economies.
It is essential to mention even from this point, that the tools provided for this first version of the project, are under development and are not exhaustively tested, so bugs and mistakes are expected to be found (and should be) so that they are fixed at a later version of the document. Additionally, it is also useful to highlight that wherever there is content generated by AI engines, they are not considered as a priori correct but are susceptible to errors and hallucinations

Relation to D3.3 and other ISABEL deliverables. D3.5 is the direct frontend counterpart to D3.3. Whilst D3.3 specifies the SNAP backend architecture, including the Synthetic Persona Generator pipeline (persona/app.py, persona/main.py), the Research Companion subsystem (research_companion/), and the unified OpenWebUI Pipe (snap_pipe), this deliverable describes how researchers interact with those components through bespoke web interfaces. The broader platform architecture, including authentication (Keycloak/OAuth 2.0), transport security (TLS via Nginx), and MinIO-backed data lake tiers, is specified in D2.5 and D5.3 respectively. All terminology in this document is aligned with D3.3 as the authoritative source of truth for SNAP component names, module paths, and architectural decisions.

The deliverable is structured as follows. Section 2 provides an overview of the researcher toolkit and its federated architecture. Sections 3, 4, 5, and 6 describe the user-facing workflows of each of the four researcher tools in detail. Section 7 addresses user interface design and workspace cohesion. Section 8 covers personalised storage and output management. Section 9 addresses validation and output trustworthiness. Section 10 concludes with a summary of researcher benefits.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

2. Overview of the Researcher Toolkit

The researcher toolkit operates as a comprehensive digital workspace that integrates four underlying computational utilities, synthetic persona simulation, academic literature discovery, unstructured data extraction, and conversational analytics, into a cohesive user journey. Rather than forcing researchers to interact with raw backend endpoints or command-line interfaces, the ecosystem provides a suite of modern, purpose-built web environments tailored specifically for complex dataset generation.

2.1. Architecture of the User Environment (Front-end focus and tool synergy)

To accommodate the diverse methodological requirements of labour market research, the user environment employs a federated frontend architecture. This approach avoids a rigid, monolithic design, instead providing researchers with distinct, highly optimised interfaces that communicate with the unified SNAP Python orchestration backend via secure REST APIs and NDJSON streaming. The three presentation layers are as follows:

1. **The Survey Simulator Interface (Vue/Nuxt):** A decoupled SPA deployed within the broader Isabel platform at the route /persona-generator. It utilises a Nuxt Nitro server proxy (server/api/persona/generate.post.ts) to eliminate Cross-Origin Resource Sharing (CORS) issues, ensuring the Flask backend URL is never exposed to the browser. It implements NDJSON streaming to render real-time progress bars and live statistics during computationally intensive LLM sampling. The component communicates exclusively with the Flask POST /generate and GET /load_saved endpoints documented in D3.3.
2. **The Research Companion Interface (Streamlit):** A highly reactive, Python-native dashboard optimised for document handling. It provides researchers with fluid toggle controls to switch between Retrieval-Augmented Generation (RAG) pipelines, manage local vector libraries, and dynamically upload PDFs for immediate academic interrogation.
3. **The Extraction & Analytics Interface (OpenWebUI):** A conversational web portal that wraps the extraction and PandasAI backend scripts. It allows researchers to use natural language chat prompts to direct document parsing and generate executable Python code for data analysis.

The synergy of the toolkit lies in its API-driven connectivity. Although the researcher interacts with different presentation frameworks depending on the task, the underlying experience remains cohesive. Complex backend processes (such as vector embeddings, ISCO-to-ESCO bridging, and strict JSON prompt engineering) are consistently abstracted behind intuitive buttons, sliders, and conversational chat boxes.

Table 1. Frontend Technology Stack Summary

Component	Framework	Port	Backend Module (D3.3)	Key Integration
Persona Generator	Vue/Nuxt (Nuxt 4)	Integrated in Isabel platform	persona/app.py (Flask, port 5002)	NDJSON streaming, Nuxt Nitro proxy
Research Companion	Streamlit	5003	research_companion/app.py	ChromaDB, Ollama

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Component	Framework	Port	Backend Module (D3.3)	Key Integration
				embeddings, vLLM
Extraction & Analytics	OpenWebUI (snap_pipe)	Integrated in OpenWebUI platform	OpenWebUI Pipe (snap_pipe)	PandasAI, OpenWebUI Knowledge Base API

2.2. Privacy-Preserving Local Execution (Ensuring data security for researchers)

A primary concern for researchers working with unreleased survey frameworks, proprietary academic literature, or sensitive demographic hypotheses is data security. Cloud-based analytical tools inherently require uploading sensitive data to third-party servers, introducing significant GDPR compliance risks and intellectual property vulnerabilities.

The SNAP user environment circumvents these risks by guaranteeing privacy-preserving local execution across all its interfaces. The entire ecosystem, spanning the Nuxt portals, Streamlit dashboards, OpenWebUI instances, Flask backends, and the underlying local LLMs (such as Ollama and vLLM), is designed to be hosted entirely within secure, air-gapped institutional networks or local machines.

Furthermore, to enhance researcher autonomy without relying on centralised cloud databases, the frontend architecture leverages local storage methodologies. The Persona Generator securely writes generated datasets and workspace states directly to the browser's localStorage as a PersonaStoredPayload schema. The Research Companion persists vector embeddings locally via ChromaDB (stored at ./data/chroma_db). The Extraction pipeline writes CSV artifacts to the host filesystem. No data ever traverses an external network boundary.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

3. Personalised Synthetic Dataset Generation (Survey Simulator)

The Synthetic Persona Generator represents a highly modular simulation workspace designed to empower researchers to curate custom microdata. Rather than operating as an unconstrained generative system that risks reproducing AI training biases, the platform implements a separation between multivariate statistical sampling and text generation. The tool is exposed to researchers through a tailored front-end workspace, allowing domain experts to actively parameterise the simulation, monitor cohort construction in real time, and export verified empirical datasets.

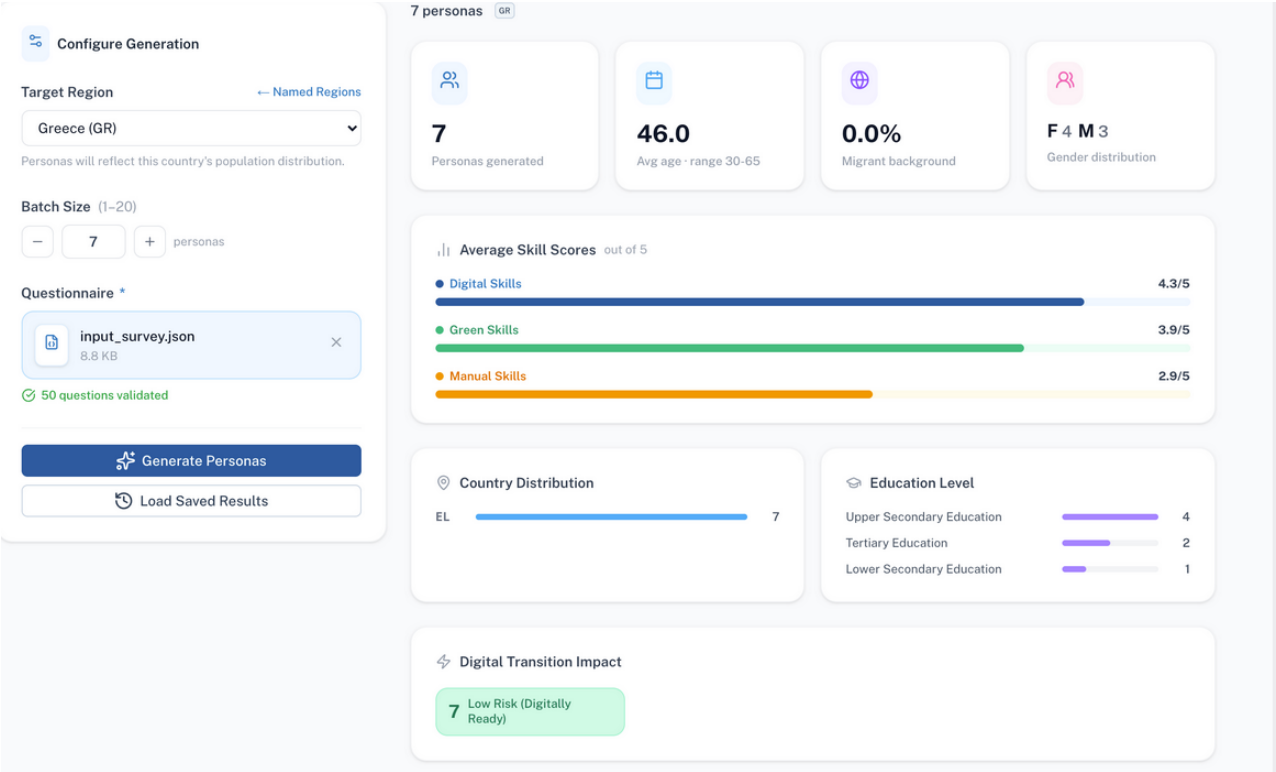


Figure 1: Configuration Panel and Macro-level Cohort Analytics. The upper section of the Survey Simulator interface displays the persistent control panel used to set geographic constraints, adjust the batch size, and upload custom questionnaires. The adjacent workspace visualises the pre-aggregated statistics, including demographic distributions and average skill scores for the generated synthetic cohort

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

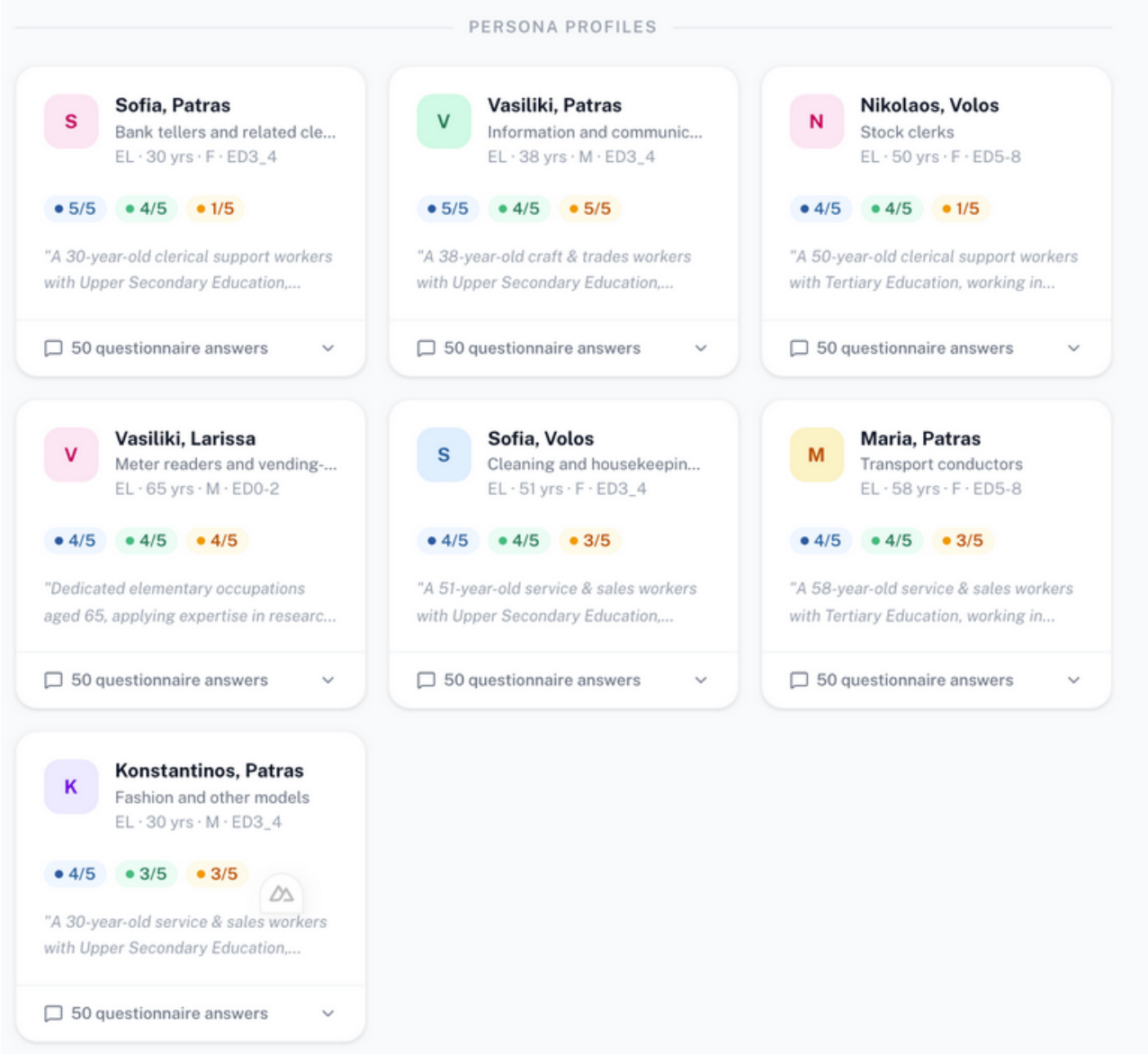


Figure 2: Synthetic Persona Profile Grid. The lower section of the workspace renders a responsive card grid that details each individual synthetic participant. Each profile card displays a deterministic colour avatar, key demographic attributes, calculated skill scores, a short biography, and an expandable section containing the structured questionnaire answers.

3.1. Custom Survey Configuration and Upload Workflows (Uploading the survey.json via UI)

The generation pipeline is configured within the persistent left-column configuration panel (PersonaControlPanel.vue) of the interface. This panel remains fixed as the researcher scrolls through results, enabling iterative re-configuration without losing the current workspace view. The workflow abstracts the complex backend statistical modelling and local LLM orchestration documented in D3.3 behind three intuitive inputs:

- **Geographic Constraints:** A labelled dropdown pre-populated with six named macro-regions (all_europe, eu27, northern, western, southern, eastern Europe) sourced from the NAMED_REGIONS constant defined in app/utils/persona.constants.ts. A toggle link switches to the full list of thirty-six individual country ISO codes

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

(COUNTRY_CODE_REGIONS), enabling precision targeting consistent with the get_allowed_countries() backend filter documented in D3.3 Section 3.1.1.

- **Batch Size Parameterisation:** A numeric stepper (− / value / +) governs the volume of the cohort (batchSize), with a default of DEFAULT_BATCH_SIZE = 5. Pressing the stepper beyond the clamped range produces no error state, guaranteeing a clean user experience.
- **Questionnaire Ingestion:** A dedicated drag-and-drop file upload zone captures the user's custom survey framework (`questionnaire`), which must be uploaded as a structured JSON file.

The Nuxt application embeds a rigorous client-side validation layer (useQuestionnaireValidation.ts) that intercepts the uploaded file immediately upon selection, prior to any API call. This prevents wasted server round-trips and provides the researcher with immediate confidence regarding the structural integrity of their survey instrument.

The validation rules enforced are as follows:

- **Structural integrity:** The file must be valid JSON and the root element must be a non-empty array.
- **Identifier uniqueness:** Every question id must be globally unique within the questionnaire.
- **Likert Scales:** Must contain an explicit scale array (e.g., [1, 2, 3, 4, 5]) to capture structured, ordinal variables.
- **Multiple-Choice Questions:** Must define a non-empty options array.
- **Free-Text Prompts:** Validated for the presence of a question string; an absent max_words field generates a warning rather than an error.

If structural errors are encountered, specific inline alerts prevent the API request from firing. Upon successful validation, the interface displays a success token (✓ N questions validated), unlocking the "Generate Personas" action button. Clicking this button triggers usePersonaState.run(), which packages the questionnaire file and configuration parameters into a multipart/form-data object and dispatches it via the Nuxt Nitro server proxy route to the backend POST /generate endpoint documented in D3.3.

Table 2. Questionnaire Question Types and Validation Rules

Question Type	Required Fields	Validation Rule
likert	id, type, question, scale	scale must be a non-empty numeric array
multiple_choice	id, type, question, options	options must be a non-empty string array
free_text	id, type, question	max_words is optional; absence triggers a warning

3.2. Live Dashboard Monitoring and Cohort Analytics (Viewing real-time generation stats)

Because generating statistically grounded personas, calculating ESCO skill matrices, and executing multi-turn LLM roleplaying are high-latency processing sequences, the UI features an active feedback layer to eliminate loading ambiguity.
The Vue/Nuxt transport layer (usePersonaApi.ts) opens a continuous chunked connection to read the incoming streaming application/x-ndjson payload. An AbortController reference enables the user to cancel mid-stream at any time via a "Cancel" button that replaces the "Generate Personas" button during generation. Partial results already streamed back are preserved and displayed with

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

an amber notice indicating the number of completed versus requested personas.
Once the final completion payload (`type: "complete"`) is received, the loading state terminates, and the right-column workspace automatically renders a comprehensive, interactive Cohort Analysis dashboard. This dashboard renders pre-aggregated statistics calculated directly by the backend server, eliminating redundant client-side sorting algorithms:

- **Demographics & Background Grid:** A responsive four-card layout displaying total participants, mean age alongside exact ranges (e.g., `41.6 · range 24–58`), gender distribution aggregates, and an empirical migrant background percentage derived from Eurostat citizenship statistics (`lfsa_pganws`).
- **Average Skill Scores Chart:** Displays three distinct horizontal progress bars mapping capability balances across labour-market skill dimensions (Digital, Green, and Manual) out of a maximum score of 5.
- **Proportional Distribution Lists:** Visualises descending list breakdowns for country-of-origin representation and ISCED education levels using horizontal inline proportional bars.
- **Digital Transition Impact Badges:** Groups the cohort by structural transition readiness levels (Low, Moderate, and High Risk), applying emerald, amber, and red semantic color coding to signify calculated reskilling vulnerabilities.

3.3. Exporting Custom Microdata (Downloading the final synthetic dataset)

Beneath the macro-level cohort analytics, the workspace renders a responsive card grid (PersonaResultsList.vue) for each individual synthetic participant. Each profile card displays:

- A deterministic colour avatar derived from the participant identifier using a 6-colour palette.
- Name, city, job title, and a demographic row (country code · age · sex · ISCED level), consistent with the NormalizedPersona model produced by usePersonaNormalization.ts.
- A migrant background badge, where applicable.
- Three skill-score pills (Digital / Green / Manual out of 5).
- A two-line clamped short biography.
- An expandable questionnaire answers section (e.g., "50 questionnaire answers ∨") rendering the structured JSON responses keyed by the original questionnaire IDs.

The data management architecture provides three persistence vectors:

1. **Automated Session Persistence:** Upon successful generation, the entire payload (analytics, normalised profiles, and summary) is written to localStorage under the key isabel_persona_last_result as a PersonaStoredPayload schema. Upon re-entering the portal, rehydrateFromStorage() restores the workspace, displaying a dismissible *"Showing results from your last session"* banner.
2. **File-Backed Retrieval:** The *"Load Saved Results"* action executes a GET /load_saved?region=<code> query via the Nuxt server proxy route (server/api/persona/load-saved.get.ts). The interface evaluates the response and re-renders historical data stored on the host filesystem (outputs/personas/*.json) by the Flask backend.
3. **Downstream Data Ingestion:** Because the Flask backend writes each validated persona directly to disk as a parseable JSON artefact (as described in D3.3 Section 3.1.6), researchers depart the portal with a structured, custom-tailored micro-dataset immediately ready for local download, offline programmatic interrogation, or integration into external statistical packages.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

4. Interactive Literature Deep-Dive (Research Companion)

The Research Companion transforms passive academic reading into an active, interactive digital workspace. Rather than simply downloading static files, researchers can use this environment to systematically discover relevant literature, construct a tailored local library, and initiate conversational question-answering loops directly with academic texts.



Figure 3: The Research Companion Discovery Workspace. The interface features a persistent sidebar where researchers establish a personalised research profile by inputting specific search keywords (e.g., "green transition" or "labour markets") and selecting the desired number of papers to retrieve. The main workspace executes the retrieval pipeline, presenting the latest fetched arXiv papers alongside their metadata, full abstracts, and concise technical summaries generated automatically by the local LLM.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

4.1. Setting up Personalised Research Profiles (Entering keywords to tailor results)

The researcher's workflow begins in the Discovery workspace (the "Daily Discovery" tab). To establish a personalised research profile, the researcher inputs specific search keywords outlining their current academic focus (e.g., "labour market transition in the EU" or "digital upskilling") in the sidebar "Search Keywords" input, and selects the desired number of papers via a slider (1–20).

Upon clicking "Fetch from arXiv", the interface invokes ResearchCompanion.fetch_papers() (documented in D3.3 Section 3.2.1), which programmatically retrieves paper metadata via the arxiv Python client. For each retrieved paper, the interface displays the core metadata (title, authors, publication date, URL) alongside the full abstract and a concise AI-generated technical summary produced by ResearchCompanion.summarize_abstract() via the remote vLLM endpoint (Mistral-Small-24B-Instruct-2501, served locally at CERTH, as specified in D3.3 Section 3.2.2)

Papers that strongly align with the researcher's hypotheses are automatically saved to the local ChromaDB collection via ResearchProfileMatcher.add_papers(). The system computes Ollama-based embeddings (nomic-embed-text:latest) for each abstract and stores them in the persistent ./data/chroma_db collection, as documented in D3.3 Section 3.2.3. The sidebar displays real-time ChromaDB statistics (total saved papers), and the "Top Matches" section renders semantically ranked results with colour-coded match score badges (🔥 >80%, ⭐ >60%, ✓ below threshold), derived from cosine similarity computed against the researcher's profile embedding.

4.2. Interacting with the RAG Pipeline (Chatting directly with specific papers)

When a researcher wishes to interrogate a specific text deeply, they transition to the "Deep Dive (RAG)" tab. This environment provides complete flexibility regarding document sourcing. Through radio button controls, the researcher may:

- **Load from the saved ChromaDB library:** A dropdown populated from matcher.collection.get() lists all previously saved papers by title, resolving the selected entry to its arXiv URL.
- **Input a direct arXiv URL:** The system invokes DeepDiveRAG.load_and_process_paper() or PaperQAEngine.load_and_process_paper_sync(), which downloads the PDF, processes it via PyMuPDFLoader, and chunks it using RecursiveCharacterTextSplitter (as described in D3.3 Section 3.2.4).
- **Upload a local PDF:** The Streamlit file uploader writes the file to a temporary path and passes it directly to the selected RAG engine, ensuring no external transmission occurs.

Once the document is loaded into active memory, a conversational chat console is unlocked. The researcher types natural language questions directly into the chat prompt. The system maintains a persistent st.session_state.chat_history list to support multi-turn conversations, ensuring the local LLM retains contextual awareness across continuous interactions.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Latest Fetched Papers

TacForeSight: Force-Guided Tactile World Model for Contact-Rich Manipulation

Authors: Yujie Zang, Yuhang Zheng, Xian Nie, Yupeng Zheng, Shuai Tian, Songen Gu, Chen Gao, Zining Wang, Shuicheng Yan, Wenchao Ding | **Date:** 2026-06-09 | URL

Local LLM Summary:

TacForeSight is a novel framework for real-time robotic manipulation that addresses the asymmetric roles of global force and local tactile sensing. It employs a tactile world model, TacForceWM, which predicts short-term tactile dynamics using dual-finger tactile observations and high-frequency wrist force signals. The framework also includes a Predictive Tactile-Conditioned Policy that leverages these predictions for adaptive, proactive contact reasoning, demonstrating superior performance in real-robot experiments, especially under dynamic contact disturbances.

View Full Abstract

Contact-rich manipulation requires robots to continuously perceive and regulate evolving physical interactions under dynamic contact transitions or complex surface geometries. Recent imitation learning methods improve contact-aware control by incorporating tactile or force feedback, but they rarely model the asymmetric spatiotemporal roles of global force and local tactile sensing. To address this, we propose TacForeSight, a lightweight force-conditioned tactile foresight framework for real-time manipulation. The core component is TacForceWM, a tactile world model that predicts short-horizon tactile latent dynamics from dual-finger tactile observations conditioned on high-frequency wrist force and torque signals. Another key component, the Predictive Tactile-Conditioned Policy, leverages the predicted latents as anticipatory contact priors, models the current-to-future tactile evolution via cross-attention, and adaptively fuses visuo-tactile features through a tactile-guided gating module. By forecasting purely within a compact latent space, TacForeSight enables proactive contact reasoning with efficient real-time inference suitable for high-frequency manipulation control. Real-robot experiments on five representative tasks and three in-process perturbation settings show that TacForeSight consistently outperforms existing baselines, particularly under dynamic contact disturbances. All models and datasets will be made publicly available on the project website at https://tacforesight.github.io/ProjectPage.

Daily Discovery & Library | Deep Dive (RAG)

Chat with PDF (100% Local)

You can choose between two RAG approaches:

Select RAG Approach:

Traditional LangChain (Original) | PaperQA (Simplified)

How would you like to load the paper?

From my saved collection (ChromaDB) | New arXiv URL | Upload local PDF

Select a paper from your collection:

An Agency-Transferring Model-Free Policy Enhancement Technique

Target URL: http://arxiv.org/abs/2606.09825v1

Load Paper into RAG Memory

How would you like to load the paper?

From my saved collection (ChromaDB) | New arXiv URL | Upload local PDF

Select a paper from your collection:

An Agency-Transferring Model-Free Policy Enhancement Technique

Target URL: http://arxiv.org/abs/2606.09825v1

Load Paper into RAG Memory

Paper successfully loaded using LangChain!

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

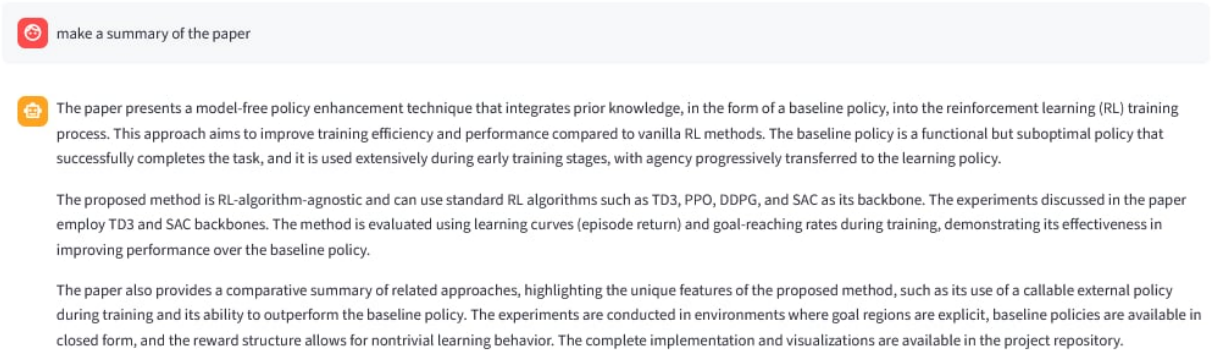


*Figure 4: Deep Dive (RAG) **Document Interrogation.** The configuration panel allows researchers to seamlessly toggle between the traditional LangChain and PaperQA RAG methodologies. Users can flexibly load academic texts into active memory by selecting papers from their saved ChromaDB collection, inputting a new arXiv URL, or uploading a local PDF. Once loaded, the conversational chat console enables researchers to submit natural language questions and receive detailed, context-aware summaries directly from the text.*

4.3. Dual RAG Mode – LangChain vs. PaperQA

Where PaperQA is installed (PAPERQA_AVAILABLE = True), the interface exposes a methodological toggle (radio button) allowing the researcher to select between the two RAG pipelines documented in D3.3 Section 3.2.5:

- Traditional LangChain RAG (phase3_deep_dive.py): Employs a DeepDiveRAG instance backed by a temporary in-memory Chroma vector store. The LCEL chain (LangChain Expression Language) uses ChatOpenAI pointed at the vLLM endpoint as the LLM, and OllamaEmbeddings for local embeddings. Retrieves the top k=15 chunks for context assembly. This mode is optimised for fine-grained, single-document interrogation with low latency.
- **PaperQA (paperqa_rag.py):** Employs a PaperQAEngine instance backed by a Docs object configured via LiteLLM wrappers. This mode provides citation-backed answer generation, tracing every assertion back to its originating text block (as described in D3.3 Sections 3.2.7–3.2.8). Responses include explicit reference lists, satisfying the strict academic requirement for source provenance

Table 3. RAG Pipeline Comparison

Characteristic	LangChain (phase3_deep_dive.py)	PaperQA (paperqa_rag.py)
Vector Store	In-memory Chroma (deleted on exit)	PaperQA Docs object (session-scoped)
LLM	ChatOpenAI → vLLM (Mistral-Small-24B)	LiteLLMModel → vLLM (vllm-mistral)
Embeddings	OllamaEmbeddings (nomic-embed-text)	LiteLLMEmbeddingModel → Ollama
Chunk retrieval	Top k=15 chunks per query	PaperQA internal retrieval
Citation support	No (context-grounded only)	Yes (explicit reference list)
Best for	Fast, exploratory single-document Q&A	Rigorous, citation-backed academic synthesis

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

4.4. Library Management and Reset

The Streamlit sidebar exposes full administrative control over the researcher's local vector library. The "Browse Saved Library" mode retrieves all records from the ChromaDB collection via matcher.collection.get(), displaying abstracts, metadata, and arXiv links in expandable accordions. A text filter allows substring-based search across titles and author names.

A dedicated "Reset Database" button invokes ResearchProfileMatcher.reset_collection(), which deletes and recreates the ChromaDB collection. This is vital for researchers managing multiple distinct projects, ensuring that semantic search results and RAG contexts are not cross-contaminated with unrelated legacy literature. Session state (latest_fetched_papers and matched_results) is simultaneously cleared, and a st.rerun() call refreshes the interface.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

5. Custom Unstructured Data Extraction

A persistent challenge in empirical labour market research is the extraction of valuable tabular data locked within unstructured document formats, such as governmental PDFs and institutional reports. The Custom Unstructured Data Extraction workspace addresses this bottleneck by providing a conversational, user-driven interface hosted within the OpenWebUI environment. Consistent with the backend architecture described in D3.3 Section 3.3, this tool is operationalised as the unified OpenWebUI Pipe module (snap_pipe).

5.1. User-Driven Document Selection and Prompting (Selecting PDFs and giving extraction prompts)

The extraction workflow is designed to operate within a unified conversational interface, granting the researcher direct control over extraction parameters. Document selection is managed through the tool's "Valves" configuration panel, accessible before initiating a chat session. The researcher inputs the unique identifiers of target documents (a comma-separated list of TARGET_FILE_IDS) for files already uploaded to and hosted within the OpenWebUI file storage system.

Once target files are defined, the researcher guides the extraction process using natural language. Typing a direct extraction query into the chat console (for example, "Extract Table XII from each file") defines the precise scope of the data required. The interface completely abstracts the aggressive prompt engineering occurring in the background, as documented in D3.3 Appendix 6.2. The system autonomously constructs a rigid instruction set that coerces the generative model to return strictly valid JSON, explicitly forbidding markdown formatting and conversational filler.

5.2. Workflow: From PDF to Personalised Knowledge Base (Getting the final structured CSV)

Immediately following the extraction request, the system executes the rigorous normalisation routine documented in D3.3 Sections 3.3.2–3.3.4:

1. **LLM-driven JSON extraction:** The Pipe invokes _call_llm_extract() with a strict system prompt mandating JSON-only output. The raw LLM response is programmatically parsed, malformed outputs are rectified, and row-to-header alignment is verified.
2. **Provenance tagging:** A _source_file_id metadata tag is automatically injected into every individual row of the extracted data. This guarantees that even when data from multiple documents is aggregated into a single table, researchers can trace every data point back to its exact originating file, satisfying academic auditing requirements.
3. **CSV generation:** The normalised data is converted to a structured Pandas DataFrame and exported as a formal CSV artefact.
4. **Knowledge Base creation:** The tool executes a sequence of authenticated POST requests to the OpenWebUI backend API (specifically targeting /api/v1/knowledge/create, /api/v1/files/, and /api/v1/knowledge/{kb_id}/file/add), uploading the CSV artefact and creating a new, personalised, vectorised Knowledge Base immediately available for semantic search.

The conversational interface provides the researcher with visual feedback detailing the number of files processed, the total row count, and the unique identifier of the newly created Knowledge Base. Static, inaccessible PDF tables are thereby transformed into interactive, vectorised datasets fully prepared for subsequent semantic queries or conversational data analytics.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

6. Conversational Data Analytics

Conversational Data Analytics represents the final stage of the researcher's workflow following data extraction. This workspace allows domain experts to interrogate structured datasets using intuitive natural language, effectively democratising advanced statistical analysis by removing the traditional barrier of manual programming in Python or R. Consistent with D3.3 Sections 3.4 and 3.4.2, the analytical agent is powered by PandasAI, integrated within the unified OpenWebUI Pipe module.

6.1. Interacting with the PandasAI Agent (Asking data questions in natural language)

The analytical workflow is seamlessly integrated into the conversational interface. Building upon the structured data tables generated in the extraction workflow (Section 5), the researcher inputs an analytical query into the chat console (for instance, "Calculate the average from both Tables").

The frontend routes this natural language prompt to the PandasAI agent (as described in D3.3 Section 3.4.1) initialised with a custom OpenWebUILLM wrapper. The previously extracted DataFrame is injected directly into the agent's contextual environment. The agent autonomously analyses the dataset's schema, data types, and column headers, translating the researcher's analytical intent into precise, executable Python data manipulation commands (as described in D3.3 Section 3.4.2). Researchers may perform complex data filtering, statistical aggregations, and comparative summaries without writing any code.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

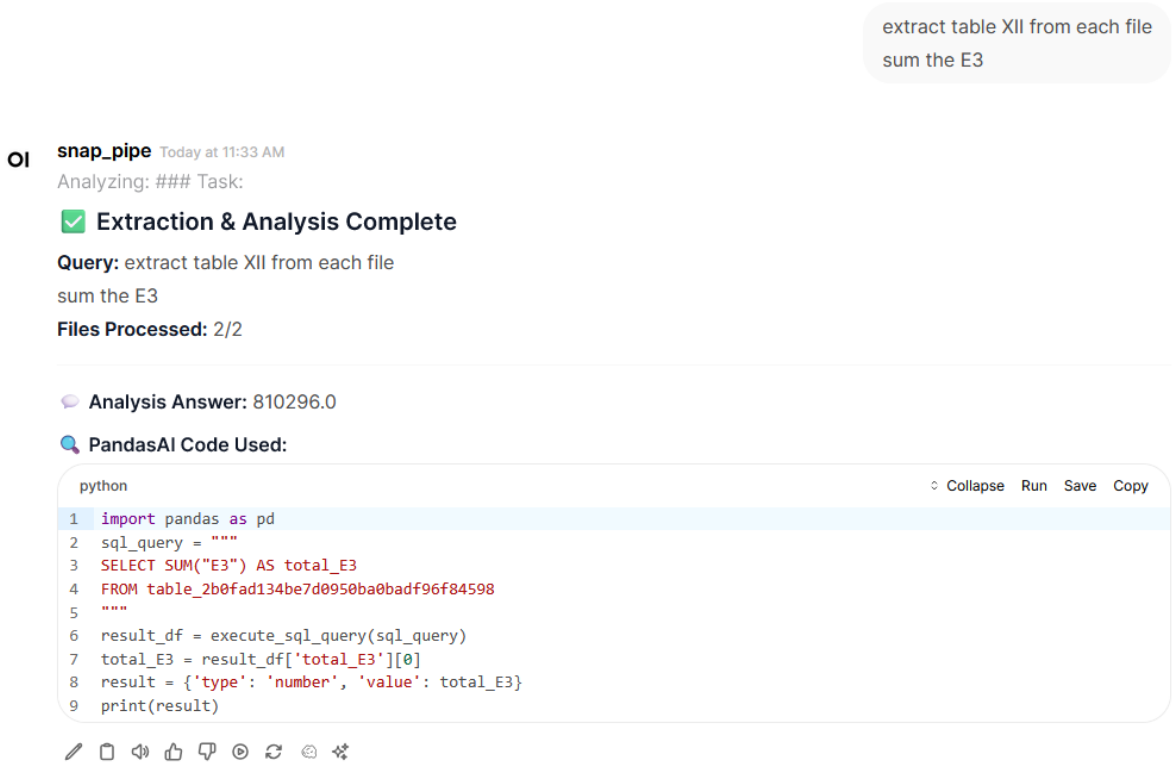


*Figure 5: The **Extraction and Analytics Interface within OpenWebUI.** The initial conversational workspace is powered by the unified snap_pipe module. Following a natural language request to extract and analyse specific tabular data, the system returns a multi-part analytical card. The interface confirms the successful document extraction, presents the synthesised statistical answer, and expands the Transparency Log to reveal the exact Python code generated and executed to derive the result.*

6.2. Verifying Generated Code and Analytical Outputs (Reviewing Python code for transparency)

A critical requirement in empirical academic research is methodological transparency. Researchers cannot rely on unverified, opaque AI calculations for their published findings. The user environment enforces complete execution transparency, consistent with D3.3 Section 3.4.4.

When the PandasAI agent completes its processing loop, the final response is rendered within the chat window as a multi-part analytical card. This card displays the synthesised statistical answer alongside a clean visual preview of the queried data. Crucially, the interface surfaces a collapsible "Transparency Log" directly beneath the answer block. When expanded, the researcher can inspect the precise programmatic syntax used during the run, exposing both the generated_code and the executed_code attributes, as described in D3.3 Section 3.4.4. By surfacing this underlying logic, the tool enables researchers to peer-review, validate, and confidently export the exact mathematical operations to their local environments, ensuring reproducibility and methodological validation for all analytically derived insights.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

7. User Interfaces and Workspace Cohesion

Despite leveraging distinct front-end technologies to serve specific methodological needs, the SNAP toolkit operates as a highly cohesive digital workspace. The overarching design philosophy prioritises researcher autonomy, ensuring that complex Python orchestration, local model inference, and vector mathematics are entirely abstracted behind intuitive, purpose-built user interfaces. This federated front-end architecture ensures that researchers experience a fluid, accessible journey from raw data ingestion to final dataset export.

7.1. The Persona Generator Interface (Vue/Nuxt)

The Persona Generator abandons traditional static dashboards in favour of a highly responsive SPA built upon the Vue/Nuxt framework, integrated into the broader Isabel platform. The interface adheres to the platform's established visual language, primary blues (consistent with the Isabel platform's design system), Public Sans typography, and the shared navigation shell (PlatformHeader, TheNavbar), ensuring seamless integration.

The interface employs a persistent two-column grid (lg:grid-cols-3). The left column (one-third width, sticky) operates as the control panel housing the region selectors, batch size steppers, and drag-and-drop questionnaire uploaders (PersonaControlPanel.vue). The right column (two-thirds width, scrollable) functions as the primary workspace, rendering in sequence: the progress indicator (PersonaProgressStatus.vue), the diagnostics panel (PersonaDiagnosticsPanel.vue), the analytics dashboard (PersonaAnalyticsCards.vue), and the persona grid (PersonaResultsList.vue). All UI components utilise UBadge, UButton, and UIcon components from @nuxt/ui for consistent interactive affordances.

The principle guiding every design decision is that "the user should never feel lost." Feedback is immediate (inline validation, pulsing progress indicators), layout is predictable (two-column grid), and the information hierarchy matches how a researcher would naturally read: high-level KPI numbers first, then demographic distributions, then individual persona detail.

7.2. The Research Companion Interface (Streamlit)

The Research Companion is deployed via a Streamlit-based environment, chosen for its robust capacity to render reactive, data-intensive Python applications without requiring a separate JavaScript frontend. The interface is explicitly structured to support intensive reading and document management workflows.

The environment utilises a persistent sidebar configuration panel, allowing researchers to adjust search parameters (keywords, number of papers) and monitor their ChromaDB library statistics globally without disrupting active reading sessions. The primary workspace is divided into two logical tabs ("Daily Discovery & Library" and "Deep Dive (RAG)"). Markdown rendering, expandable text accordions for abstracts, and integrated file uploaders provide a clean, document-centric workspace that feels immediately familiar to academic researchers. Colour-coded Match Score badges provide rapid visual differentiation of paper relevance

7.3. The Extraction and Analytics Interface (OpenWebUI)

To handle the highly variable nature of unstructured data extraction and programmatic analysis,

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

the toolkit leverages the conversational interface paradigm provided by OpenWebUI. Rather than forcing researchers to navigate complex menus, the environment mimics a collaborative chat application.

Researchers configure their target documents via integrated Valves settings panels and then dictate extraction or analytical queries directly into the primary chat input field. The interface excels at rendering diverse data types within the conversational thread: when a table is extracted, the UI renders a clean Markdown data grid directly within the chat bubble; when a PandasAI query is executed, the interface displays the statistical answer alongside a collapsible Diagnostics Drawer exposing the generated Python code. This conversational approach unifies extraction, structuring, and analysis into a single, flowing interaction.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

8. Personalised Storage and Output Management

The creation of custom datasets represents only the first half of the research lifecycle. The subsequent storage, retrieval, and systematic management of these digital assets are equally critical. The SNAP toolkit is engineered to function as a persistent digital workspace, ensuring that every generated demographic cohort, extracted data table, and academic summary is meticulously organised and securely preserved. By eschewing ephemeral cloud storage in favour of local, file-backed persistence, the user environment provides researchers with absolute ownership and sovereign control over their empirical outputs.

8.1. Local Vector Profiles (Managing ChromaDB user collections)

Within the Research Companion workspace, the management of accumulated academic literature is handled through the curation of local vector profiles. As researchers ingest, evaluate, and save papers during their discovery sessions, the system autonomously constructs a dense semantic memory using a local ChromaDB instance.
Through the user interface, researchers exercise full administrative control over this vectorised library. The settings panel surfaces real-time database statistics, allowing users to monitor the total volume of saved papers. Furthermore, researchers can actively browse, search, and filter this repository to retrieve specific historical abstracts or metadata.
Afterwards, the interface provides a dedicated "Reset Database" functionality. This feature is vital for researchers managing multiple distinct projects, it allows them to instantly clear their current vector collection and construct an entirely new, cleanly separated research profile. By resetting the database, researchers ensure that when they shift their academic focus (for instance, from "green energy policies" to "digital platform labour") their semantic search results and RAG contexts are not cross-pollinated with unrelated legacy literature.

8.2. Managing Exported Datasets and CSVs (Organizing final outputs locally)

For quantitative and unstructured data workflows, the system generates tangible, highly structured files that require rigorous local organisation. When a synthetic demographic cohort is generated via the Survey Simulator, the complete schema, including all parameter configurations, normalised profile attributes, and LLM-generated survey responses, is automatically written to the host filesystem as a structured JSON artifact. This file-backed persistence is what powers the interface's "Load Saved Results" functionality, allowing researchers to effortlessly retrieve and review historical simulation runs.
Similarly, within the extraction workspace, when tabular data is liberated from unstructured governmental or institutional reports, the interface automatically converts the normalised data grids into formal CSV files. These CSV artifacts serve a dual purpose. First, they are dynamically injected into the system's active file registry to construct new Knowledge Bases, enabling immediate conversational analysis via PandasAI. Second, they are explicitly organised and saved within the local directory structure.
This systematic output management guarantees that researchers conclude their interactive sessions possessing pristine, highly organised datasets. These final artifacts are entirely decoupled from the SNAP ecosystem, meaning they are immediately ready to be ingested into external statistical software packages (such as R, SPSS, or Stata), shared securely with institutional collaborators, or attached as verifiable empirical appendices to formal academic publications.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

9. Validation and Output Trustworthiness

The scientific utility of synthetic microdata and AI-assisted analytics relies entirely upon the researcher's ability to trust the generated outputs. If the user environment acts as an opaque "black box," the resulting datasets cannot be confidently deployed in rigorous academic or policy-driven research. Therefore, the SNAP toolkit embeds comprehensive validation mechanisms directly into the presentation layer, ensuring that researchers can continuously audit the trustworthiness of their personalised datasets.

9.1. Statistical Representativeness and Error Transparency in the Persona Generator

When researchers utilise the Survey Simulator to generate a custom demographic cohort, they must be certain that the resulting sample is statistically representative of the target region and free from unmeasured AI generation errors. The user interface achieves this through aggressive execution transparency and detailed error surfacing.

Because the underlying orchestration relies on LLM roleplaying and strict JSON coercion (as described in D3.3 Section 3.1.6), individual persona generations may occasionally fail. Rather than silently masking these errors, the frontend (PersonaDiagnosticsPanel.vue) is designed to expose partial successes. If a batch completes with dropped records, the interface renders a "Diagnostics Panel" displaying a prominent amber warning explicitly contrasting the number of successfully generated personas against the originally requested batch size. A collapsible diagnostic log lists the exact system errors or prompt failures. By surfacing these metrics, the interface empowers the researcher to calculate the statistical attrition rate of their synthetic sample and determine whether simulation parameters require adjustment.

9.2. Citation Provenance in RAG Outputs

Within the Research Companion's PaperQA mode, the citational provenance of every generated assertion is preserved throughout the pipeline (as described in D3.3 Sections 3.2.7–3.2.8). Responses returned to the researcher include explicit citation markers traceable to originating text blocks within the source PDFs. This guarantees that all derived literature reviews maintain absolute methodological transparency and satisfy the strict academic requirement that LLM-synthesised arguments are verifiable against original source materials.

9.3. Code Logging and Analytical Reproducibility

Within the Conversational Analytics workspace, the PandasAI Transparency Log (described in Section 6.2 and D3.3 Section 3.4.4) exposes both the generated_code and executed_code for every analytical query. This enables researchers to peer-review the exact programmatic logic driving their findings, export the code for independent verification in external environments, and document the precise computational steps in their academic publications. This capability directly addresses the Horizon Europe requirement for open, reproducible research methodology.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

10. Conclusion (Summary of researcher benefits)

Deliverable 3.5 formalises the researcher-facing dimension of the ISABEL project's data management framework. While previous technical deliverables established the rigorous backend orchestration required to fuse Eurostat statistics with ESCO taxonomies, this document demonstrates how those complex pipelines are operationalised into an accessible, human-in-the-loop digital workspace.

The custom-tailored researcher toolkit successfully abstracts the steep technical barriers traditionally associated with machine learning, vector databases, and programmatic data extraction. By providing a cohesive suite of modern web interfaces—the Vue/Nuxt Persona Generator integrated within the Isabel platform, the Streamlit Research Companion, and the OpenWebUI conversational analytics environment—the toolkit ensures that domain experts can interact with sophisticated AI pipelines using intuitive controls, file uploads, and natural language. All three components are strictly consistent with the SNAP backend architecture specified in D3.3, communicating exclusively with the locally hosted services documented therein.

The primary benefits delivered to the European research community through this toolkit can be summarised in three core pillars:

1. **Research Autonomy:** Researchers are no longer dependent on data engineers to parse unstructured reports or write complex Python analysis scripts. The toolkit allows them to autonomously configure, generate, extract, and interrogate highly personalised datasets on demand, using intuitive graphical interfaces and natural language.
2. **Absolute Data Sovereignty:** By enforcing a local-first execution paradigm and utilising browser-based localStorage alongside host-level file directories, the environment guarantees that proprietary academic literature and sensitive demographic hypotheses never leave the secure institutional perimeter. This is consistent with the GDPR-compliant, local-first model orchestration paradigm described in D3.3 Section 5.1, D2.5, and D5.3

3. **Methodological Transparency and Trust:** Every subsystem is engineered to combat AI opacity. Whether it is the provision of exact Python code logs in the PandasAI Transparency Log, the interactive citation markers in the PaperQA RAG workspace, the inline questionnaire validation in the Persona Generator, or the explicit partial-success error surfacing in the Diagnostics Panel, researchers are provided with the complete methodological provenance required to validate their findings.

Ultimately, the SNAP researcher toolkit provides a secure, flexible, and methodologically transparent infrastructure. It equips researchers with the exact precision required to generate bespoke microdata, thereby accelerating critical analysis regarding the uneven socio-economic impacts of the European transition towards green and digital economies.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Bibliography

ESCO: European Commission. (n.d.). European Skills, Competences, Qualifications and Occupations (ESCO). Directorate-General for Employment, Social Affairs and Inclusion. Retrieved from https://esco.ec.europa.eu/
ISCO: International Labour Organization (ILO). (2012). International Standard Classification of Occupations: ISCO-08. Geneva: ILO.
Eurostat LFS: Eurostat. (n.d.). European Union Labour Force Survey (EU-LFS). Datasets referenced: lfsa_egised, lfsa_pganws, lfsa_eiisco. Retrieved from https://ec.europa.eu/eurostat/
Streamlit: Streamlit Inc. (2024). Streamlit: The fastest way to build and share data apps. Retrieved from https://streamlit.io/
OpenWebUI: Open WebUI. (2024). Open WebUI: User-friendly WebUI for LLMs. Retrieved from https://openwebui.com/
vLLM: Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., ... & Stoica, I. (2023). Efficient Memory Management for Large Language Model Serving with PagedAttention. In Proceedings of the 29th Symposium on Operating Systems Principles (SOSP '23).
Mistral: Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., ... & Sayed, W. E. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.
Ollama: Ollama. (2024). Ollama: Get up and running with large language models locally. Retrieved from https://ollama.com/
PaperQA: Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., White, A. D., & Schwallko, P. (2023). PaperQA: Retrieval-Augmented Generative Agent for Scientific Research. arXiv preprint arXiv:2312.07559.
LangChain: Chase, H. (2022). LangChain. Retrieved from https://github.com/langchain-ai/langchain
PandasAI: PandasAI. (2024). PandasAI: Conversational Data Analysis. Retrieved from https://github.com/Sinaptik-AI/pandas-ai
ChromaDB: Chroma. (2024). Chroma: The AI-native open-source embedding database. Retrieved from https://trychroma.com/
SentenceTransformers: Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.
GDPR: European Parliament and of the Council. (2016). Regulation (EU) 2016/679 – General Data Protection Regulation. Official Journal of the European Union, L 119, 1–88.

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union

Annexes

Annex A: Example Questionnaire JSON Schema

The following is a representative example of the questionnaire JSON schema accepted by the Persona Generator's client-side validation layer (useQuestionnaireValidation.ts) and the Flask POST /generate endpoint. This schema is consistent with persona/input_survey_v2.json as documented in D3.3 Appendix 6.1:

```
[
  { "id": "q1", "type": "likert", "scale": [1,2,3,4,5], "question": "How comfortable are
you with adopting new digital tools at work?" },
  { "id": "q2", "type": "free_text", "max_words": 50, "question": "Describe your primary
concern regarding the green transition in your sector." },
  { "id": "q3", "type": "multiple_choice", "options": ["Yes","No","Not sure"], "question":
"Do you believe that the green transition will create more job opportunities in your
field?" }
]
```

The ISABEL project has received funding from the European Union's Horizon research and innovation actions program under grant agreement No 101177687

Funded by the European Union