UPTOOL
General specifications

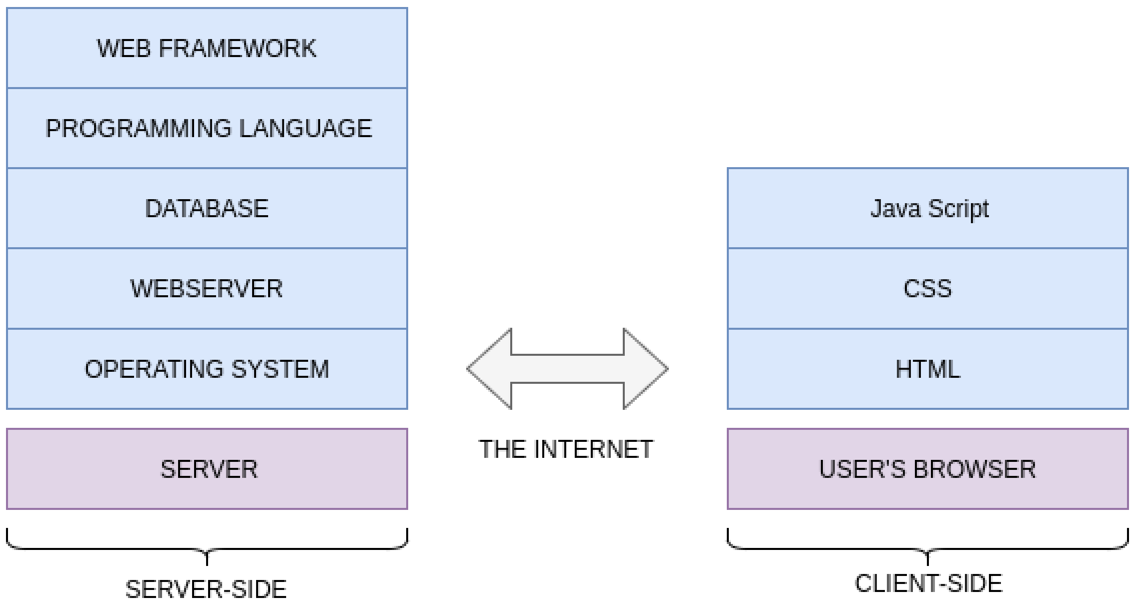
Co-funded by the Erasmus+ Programme of the European Union

UPTOOL Portal Global architecture

1. Technological choices

A tech stack is a combination of software products and programming languages used to create a web or mobile application. Applications have two software components: client-side and server-side, also known as front-end and back-end.

Each layer of the application builds on the features of the one below it, creating a stack. This diagram shows the major building blocks of the recommended tech stack, but there can be other supporting components included.



1.1. Back-end tech stack

The back-end contains the business logic that works behind the scenes to drive your application. Users will never directly engage with the back-end, all information is passed back and forth through the front-end.
The recommended server side programming language is PHP.
PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language.

1.1.1. LAMP stack

LAMP is an archetypal model of web service stacks, named as an acronym of the names of its original four open-source components: the Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language. The LAMP components are largely interchangeable and not limited to the original selection. As a solution stack, LAMP is suitable for building dynamic web sites and web applications.

Since its creation, the LAMP model has been adapted to other componentry, though typically consisting of free and open-source software.

1.1.2. Server Side Programming Language

As compared to the other programming languages, PHP is rather easy to learn and maintain. This is mainly because its syntax is based on the languages such as C and Perl.

The availability of multiple frameworks in PHP helps us to quickly build applications that incorporate a variety of different features. These frameworks also impart security and speed to our application development process.

1.1.3. Client Side Programming Language

JavaScript is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production; the majority of websites employ it, and all modern Web browsers support it without the need for plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

1.1.4. Framework

The recommended PHP framework is CodeIgniter.

CodeIgniter is a PHP full-stack web framework that is light, fast, flexible and secure.

It is an Application Development Framework - a toolkit - for people who build websites using PHP. Its goal is to enable you to develop projects much faster than you could if you were writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries.
CodeIgniter lets you creatively focus on your project by minimizing the amount of code needed for a given task.

Where possible, CodeIgniter has been kept as flexible as possible, allowing you to work in the way you want, not being forced into working any certain way. The framework can have core parts easily extended or completely replaced to make the system work the way you need it to.

In short, CodeIgniter is the malleable framework that tries to provide the tools you need while staying out of the way.

1.1.5. API

The recommended way of developing the UPTOOL application is by separating the application programming interface from the presentation layer.

In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it's a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

1.2. Front-end tech stack

The front-end is the visual part of your application that users will see and interact with. This interaction can happen through a web browser or a mobile app. When building for the web, the front-end tech stack is made up of:

- HTML (Markup Language)
- CSS (Style Sheet Language)
- JavaScript (Scripting Language)

1.2.1. HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard.

1.2.2. CSS3

CSS3 is the latest evolution of the Cascading Style Sheets language and aims at extending CSS2.1. It brings a lot of long-awaited novelties, like rounded corners, shadows, gradients, transitions or animations, as well as new layouts like multi-columns, flexible box or grid layouts.

1.2.3. Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Unlike many web frameworks, it concerns itself with front-end development only. Bootstrap Is free, open-source and MIT licenced.

1.2.4. Typescript

TypeScript is a superset of JavaScript which primarily provides optional static typing, classes and interfaces. One of the big benefits is to enable IDEs to provide a richer environment for spotting common errors as you type the code.

1.3. Underlying Architecture

1.3.1. Microservices

Microservices is a specialisation of an implementation approach for service-oriented architectures used to build flexible, independently deployable software systems.
Services in a microservice architecture are processes that communicate with each other over a network in order to fulfil a goal. These services use technology-agnostic protocols.

In a microservices architecture, services should have a small granularity and the protocols should be lightweight. The benefit of distributing different responsibilities of the system into different smaller services is that it enhances the cohesion and decreases the coupling.
This makes it easier to change and add functions and qualities to the system at any time.
It also allows the architecture of an individual service to emerge through continuous refactoring, and hence reduces the need for a big up-front design and allows for releasing software early and continuously.

1.3.2. Operating System

The recommendations for the underlying operating system is currently Ubuntu.

Ubuntu is built on Debian's architecture and infrastructure. Ubuntu releases updated versions predictably every six months, and each release receives free support for nine months with security fixes, high-impact bug fixes and conservative, substantially beneficial low-risk bug fixes.

1.3.3. Data Storage

Data Storage is usually broken down into two categories :

- Relational or non-relational databases
- File Storage for user uploads or application generated files (cache, logs, exports…)

Relational Databases

Relational data such as user accounts need to be stored in a relational database such as MySQL (MariaDB) or PostgresSQL, official Docker containers exist for both.
MariaDB is a community-developed fork of the MySQL relational database management system intended to remain free under the GNU GPL. Development is led by the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle Corporation.
PostgreSQL, often simply Postgres, is an object-relational database, with additional "object" features – with an emphasis on extensibility and standards compliance. As a database server, its primary functions are to store data securely and return that data in response to requests from other software applications. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

Non Relational Databases

The tracked information, however it will be tracked, needs to be stored. This information will typically use formats that cannot be known in advance and need to be stored in a non relational database that supports flexible JSON like documents.

The currently recommended technology is MongoDB.

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc. and is free and open-source, published under a combination of the GNU Affero General Public License and the Apache License.

1.3.4. Reverse proxy and SSL

HTTPS must be used in all public facing web servers to guarantee user security and to avoid potential Man in the Middle attacks.

Let’s Encrypt is a free, automated, and open certificate authority brought to you by the non-profit Internet Security Research Group (ISRG).

2. Methodological choices

To manage the product development, the framework will be based on the agile SCRUM method. Scrum is an iterative and incremental agile software development framework. It implies specific roles and workflow.

2.1. Roles

There are three core roles in the Scrum framework : the Scrum Master, the Product Owner and the development team.

1. Scrum Master: their role is to remove obstacles to allow the developer and Product owner to work as efficiently as possible. He ensures that the framework is followed.
→ sensed persons: Donovan, Philippe

2. Product Owner: plays the role of the final user and answers questions for development choices. The delay of answer from the product owner to Scrum Master or IT Developers : 24 h
→ sensed persons: Pauline (Claroline), Simon, AGH

3. Development team
→ IT developers at Haikara and Claroline

2.2. Workflow

The sprint, a period of one month or less is the heart of the Scrum framework. During this period useable and potentially releasable product increment is created. A new sprint starts immediately after the conclusion of the previous Sprint.
Each Sprint starts with a *Sprint Planning event* that aims to define the work for the Sprint and make an estimated commitment for the Sprint goal. In other words, it’s during this Sprint Planning that is decided the features to be developed during the sprint.
The sprint ends with a *Sprint Review* that reviews progress to show to stakeholders and identify lessons and improvements for the next Sprints.

3. UPTOOL ARCHITECTURE

3.1. Objective

UPTOOL aims to track learners’ activities on different types of learning content to see which one helps them learn best.

3.2. Principle

Three learning topics have been chosen:
Digital Literacy
Media Literacy
Math Literacy

Three types of learning content have been produced for all three above topics:
Text / Image
Video
Interactive content

All of this learning content is hosted by haikara and two of the project partners on different Learning Management Systems (LMS).

So we had to find a way to collect data (learners’ activities) whatever the LMS is.

3.3. Learning activity Database

The goal is to have an independent database to store all the data coming from different LMS.

The obvious choice to do this is to use the Elearning industry standard: xAPI.

3.3.1. xAPI for Learning Management Systems (LMS)

So that a third party application can communicate with the UPTOOL platform, an exchange format/protocol must be defined. The recommended exchange format for a modern web application is JSON.
On the UPTOOL side, JSON must be consumed. The UPTOOL side does not need to send information to the LMS.
On the LMS side, JSON must be sent with either JSON or Form encoding.

We have first searched for existing exchange formats/protocols.
We looked at two standards used :

- Learning Tools Interoperability (LTI)
- Tin Can API /xAPI

3.3.1.1. Previous research about existing standards

LTI
We can exclude LTI since its primary purpose is to connect learning systems with external service tools in a standard way across learning systems.

Tin Can API / xAPI
TinCanAPI / xAPI is a technical specification that aims at facilitating the documentation and communication of learning experiences. It allows us to communicate statement objects to a LRS (Learning Record Store). The exchange format is JSON. Using xAPI implies that UPTOOL will need a LRS. The xAPI data model is the "Statement" object : the principal properties of the Statement object are : "Actor", "Verb", "Object", "Result", and "Context".

3.3.1.2. TinCan API / xAPI

The Experience API (xAPI), also known as the Tin Can API, is an e-learning software specification that allows learning content and learning systems to speak to each other in a manner that records and tracks all types of learning experiences. Learning experiences are recorded in a Learning Record Store (LRS).

The exchange format is JSON. The xAPI data model is the "Statement" object : the principal properties of the Statement object are : "Actor", "Verb", "Object", "Result", and "Context".

Statement

A data structure showing evidence for any sort of experience or event which is to be tracked in xAPI as a Learning Record. A set of several Statements, each representing an event in time, may be used to track complete details about a learning experience. Requirements :

- Statement and other objects SHOULD NOT include properties with a value of an empty object.
- Statement MUST use each property no more than one time

- If Object is an Activity:
- A Statement MUST use "actor", "verb", and "object". The other properties are optional.
- A Statement MAY use its properties in any order.

Statements are information about a tracked learning experience. Typically, the information represented in the Statement has already happened. Statements are expected to be permanent. The only way to undo a Statement within this specification is to void it. Voiding does not destroy a Statement, rather indicates the evidence in the Statement is to be disregarded.

Actor

The Actor defines who performed the action. The Actor of a Statement can be an Agent or a Group.

- If Agent (persona or system):
  - An Agent MUST be identified by one of the four types of Inverse Functional Identifiers (mbox/mbox_sha1sum/openid/account - *A user account on an existing system e.g. an LMS or intranet*-)
  - An Agent MUST NOT include more than one Inverse Functional Identifier;
  - An Agent SHOULD NOT use Inverse Functional Identifiers that are also used as a Group identifier.

Property	**Type**	**Description**	**Required**
Object Type	string	Agent. This property is optional except when the Agent is used as a Statement's object.	Optional
Name	string	Full name of the Agent.	Optional
IFI		mbox or mbox_sha1sum or openid or account	Required

- If Group:

Property	**Type**	**Description**	**Required**
Object Type	string	Agent. This property is optional except when the Agent is used.	Optional
Name	string	Name of the group	Optional
Member	Array of Agent Objects	The members of this Group. This is an unordered list.	Required
IFI			Required if not anonymous group

Verb

The Verb defines the action between an Actor and an Activity. The xAPI does not specify any particular Verbs except for "voided"

Property	**Type**	**Description**	**Required**
id	IRI	Corresponds to a Verb definition. Each Verb definition corresponds to the meaning of a Verb, not the word.	Required
display	Language Map	The human readable representation of the Verb in one or more languages	Optional

Object

The Object defines the thing that was acted on. The Object of a Statement can be an Activity ("Jeff wrote an essay about hiking."), Agent/Group ("Nellie interviewed Jeff") , SubStatement, or Statement Reference ("Nellie commented on 'Jeff wrote an essay about hiking.").

Property	**Type**	**Description**	**Required**
Object Type	string	MUST be Activity when present	Optional
id	IRI	An identifier for a single unique Activity	Required
definition	Object	Object with Name, description, type (IRI), interaction (object),extensions (object)	Optional

- If Object is an Agent/Group:

Statements that specify an Agent or Group as an Object MUST specify an "objectType" property.

- If Object is a Statement:

There are two possibilities for using a Statement as an Object. First, an Object can take on the form of a Statement that already exists by using a Statement Reference. A common use case for Statement References is grading or commenting on an experience that could be tracked as an independent event. The special case of voiding a Statement would also use a Statement Reference. Second, an Object can be a brand new Statement by using a SubStatement.

- Statement references : is a pointer to another pre-existing Statement

Property	**Type**	**Description**	**Required**
Object Type	string	In this case, MUST be StatementRef.	Required

id	UUID	The UUID of a Statement.	Required

- Substatement reference: One interesting use of SubStatements is in creating Statements of intention. For example, using SubStatements we can create Statements of the form " ( )" to indicate that we've planned to take some action. A SubStatement MUST specify an "objectType" property with the value SubStatement. He MUST be validated as a Statement in addition to other SubStatement requirements (MUST NOT have the "id", "stored", "version" or "authority" properties and MUST NOT contain a SubStatement of its own).

Other properties

- Result

An optional property that represents a measured outcome related to the Statement in which it is included.

Property	**Type**	**Description**	**Required**
Score	Object	Outcome of a graded Activity achieved by an Agent. Must be scaled (Decimal number between -1 and 1, inclusive), may have a raw, max and min	Optional
Success	Boolean	Indicates whether or not the attempt on the Activity was successful.	Optional
Completion	Boolean	Indicates whether or not the Activity was completed.	Optional
Response	String	A response appropriately formatted for the given Activity.	Optional
Duration	Duration	Period of time over which the Statement occurred. MUST be expressed using the format for Duration in ISO 8601:2004	Optional
Extensions	Object	A map of other properties as needed.	Optional

- Context

Context that gives the Statement more meaning. Examples: a team the Actor is working with, altitude at which a scenario was attempted in a flight simulator.

- Timestamp

Timestamp of when the events described within this Statement occurred. Set by the LRS if not provided. The "timestamp" property is of type Timestamp. It is formatted according to the normal format of ISO 8601 and corresponds to the time of when the events described within this Statement occurred. If it is not included in the Statement when it is submitted to the LRS, the LRS populates it with the same value it would use with Stored.

- Authority

Agent or Group who is asserting this Statement is true. Verified by the LRS based on authentication. Set by LRS if not provided or if a strong trust relationship between the Learning Record Provider and LRS has not been established.

Learning Record Store (LRS)

A Learning Record Store is a database designed specifically for storing learning activity data. As we will be creating data with the Experience API (xAPI or Tin Can) specification, we will need a Learning Record Store to store, sort and share that data.

The UPTOOL content will create statements. Once captured, this data needs to be correctly stored.If the data we collect is stored incorrectly or not valid to the xAPI standard, we will find ourselves with a whole lot of useless data in years to come.
We searched for different open source LRS :

- Learning Locker: https://github.com/LearningLocker/learninglocker
- LXHIVE: https://github.com/Brightcookie/lxHive
- TRAX LRS: https://traxlrs.com/

In our case the recommended Learning Record Store is TRAX since it's based on the same technologies as the ones recommended for the UPTOOL application, and very up to date.
TRAX can collect data from Learning Management Systems, Mobile Apps, Portals, Devices and more. It scales massively and keeps data consistent with the xAPI.
Learning Locker is licensed under GPL 3.0.

We can imagine a query from UPTOOL to LRS API for each user.

You can filter the statements by :

- Who? (actor)
- Did what? (verb)
- To what? (object)
- In what? (platform, instructor, parent activity, grouping activity, language)
- With what? (result)
- When?

3.3.2. UPTOOL Statement Format

The different possibilities given by the xAPI statement format underlines the necessity of a specific UPTOOL statement format. Here is an example of a statement stored in the UPTOOL LRS:

```
{
  "id": "6716b7e7-9726-4cee-8544-15d7e701f5d1",
  "actor": {
    "name": "UPTOOL",
    "mbox": "mailto:80b7a1e3fae7f467d9e38e0e3937503f@uptool.formavisa.eu"
  },
  "verb": {
    "id": "http://digital",
    "display": {
      "en-US": "experienced"
    }
  },
  "object": {
    "id": "http://example.com/activities/solo-hang-gliding",
    "definition": {
      "name": {
        "en-US":
"{\"hostname\":\"uptool.formavisa.eu\",\"lesson_name\":\"digital\",\"lesson_level\":1,\"lesson_lang\":\"fr\",
\"slides\":{\"Titre\":{\"duration\":{\"total\":0,\"active\":0,\"inactive\":0},\"tab_exit\":0,\"video\":{\"pause\":0,
\"fast_forward\":0,\"subtitle\":false}},\"Objectif\":{\"duration\":{\"total\":6,\"active\":6,\"inactive\":0},\"tab_
exit\":0,\"video\":{\"pause\":0,\"fast_forward\":0,\"subtitle\":false}},\"Sommaire\":{\"duration\":{\"total\":38
,\"active\":38,\"inactive\":0},\"tab_exit\":0,\"video\":{\"pause\":0,\"fast_forward\":0,\"subtitle\":false}}},\"nu
mber_of_connections\":1,\"number_of_connections_before_completion\":1,\"number_of_connections_after
_completion\":0,\"tab_exit\":0,\"sessions\":[{\"progression\":[\"Titre\",\"Objectif\",\"Sommaire\"],\"duration
\":44}],\"duration\":{\"active\":44,\"inactive\":0,\"total\":44},\"average_time_to_action\":14.6666666666666
66,\"number_of_actions\":3,\"user_agent\":\"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0
Safari/537.36\",\"browser\":\"Chrome\",\"device\":\"desktop\",\"last_slide\":\"Sommaire\",\"user_id\":\"80b
7a1e3fae7f467d9e38e0e3937503f\",\"user_email\":\"80b7a1e3fae7f467d9e38e0e3937503f@uptool.formavis
a.eu\",\"quizBefore\":[],\"quizAfter\":[],\"satisfactionQuestionnaire\":[]}"
      }
    }
  },
  "timestamp": "2024-11-21T08:47:08.9537Z",
  "stored": "2024-11-21T08:47:08.9537Z",
  "authority": {
    "objectType": "Agent",
    "account": {
      "name": "authority",
      "homePage": "http://traxlrs.com"
    }
  },
  "version": "1.0.0"
```

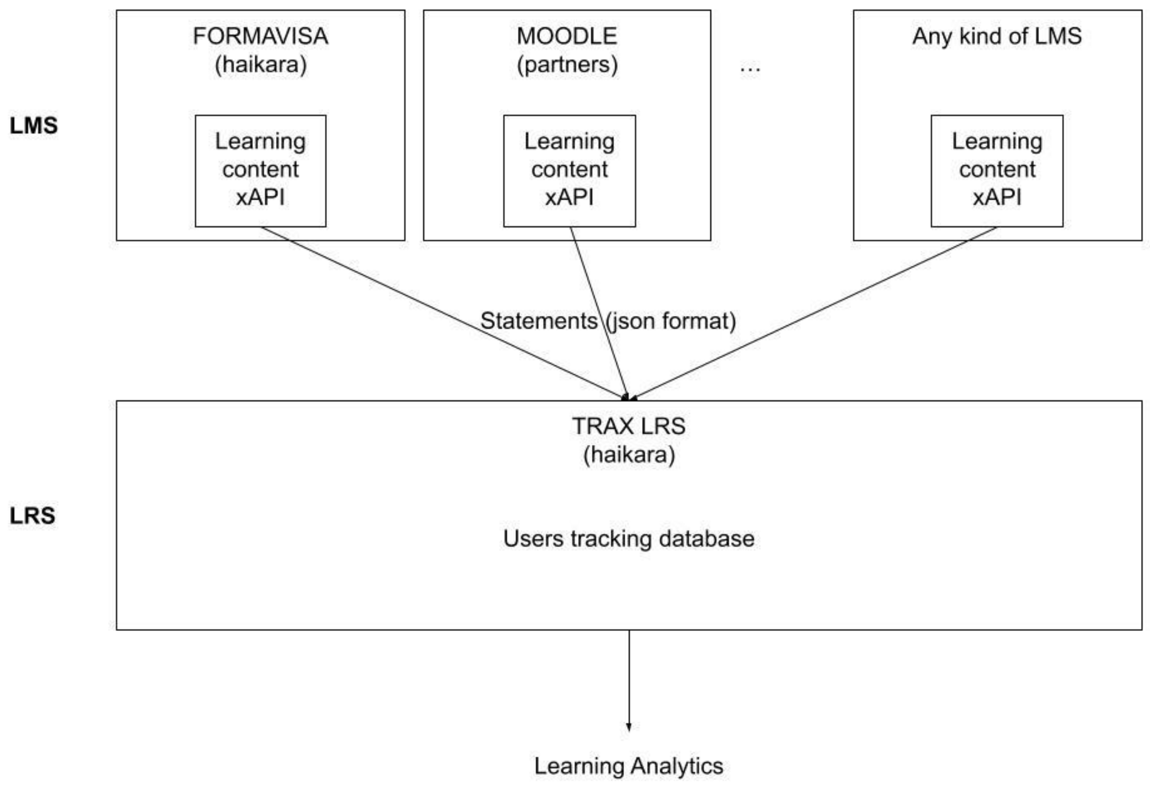
}
Here is an explanation of the data contained in the statement:

Parameter	Description	Example
name	The name of the module (maths, media or digital).	
level	The level of the module, between 1 and 3.	
slides	The name of each screen viewed is recorded. The time spent on each screen is calculated in seconds. This time spent is the sum of the active and inactive time of the tab. The inactive time corresponds to the time when the module is not visible. Active time corresponds to the time when the module is visible. Tab changes are counted, as well as the number of pauses and the activation of subtitles if the slideshow is a video.	{ "Medienkompetenz": { "duration": { "total": 760, "active": 611, "inactive": 149 }, "tab_exit": 2, "video": { "pause": 0, "subtitle": false } } }
number_of_connections	Number of connections to the module.	
number_of_connections_before_completion	Number of connections to the module before it is 100% complete.	
number_of_connections_after_completion	Number of connections to the module after it is 100% complete.	
tab_exit	Sum of all browser tab changes when viewing the module.	
sessions	Each screen is recorded in a table in the order in which it was viewed.	["Medienkompetenz","Willkommen","Einleitung","Klassische und neue Medien","Klassische Medien"]
duration_active	Time in seconds that the module is active (in the foreground).	
duration_inactive	Time in seconds that the	

	module is inactive (tab hidden or browser not in foreground).	
duration_total	Sum of active and inactive time in seconds.	
average_time_to_action	Average time between each action (action: mouse click or keyboard event).	
number_of_actions	Number of mouse clicks and number of keyboard events.	
user_agent	The user agent sent by the browser used. This tells you which browser is being used.	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0 Mozilla Firefox (Firefox/115.0)
browser	Browser used.	Edge (chromium based) Chrome Mozilla Firefox Safari
device	Medium used (computer or mobile device).	Desktop /mobile
last_slide	Name of the last screen viewed (before leaving the module).	
quiz_before_1 quiz_before_2	Answer given by the user for each question in the upstream quiz.	
quiz_after_1 quiz_after_2	Answer given by the user for each question in the downstream quiz.	
quiz_score_before	Upstream quiz score.	
quiz_score_after	Downstream quiz score.	
satisfaction_questionnaire	Answers given by the user to the satisfaction questionnaire.	
lang	Module language.	fr, de, en
date stored	date of last statement.	
profiling_1 to 10_3	Answers given by the user to the ‘Tell us a little about yourself’ questionnaire	

date_stored_profiling	profiling response date	

3.3.3. To summarise



https://uptool.formavisa.eu/
https://lrs.harkhan.com/

FORMAVISA	TRAX LRS
● Apache/2.4.25 (Debian) ● Database MariaDB version 10.1.26 ● phpMyAdmin 4.2.12 ● PHP 7.0.30 ● Codeigniter 3.1.6	● Ubuntu Server 22.04 LTS "Jammy Jellyfish" ● MariaDB 10.6.18 ● PHP 8.2.24

3.3.4. Data transfer protocol

In order to have no dependencies from any kind of LMS, we added software development directly inside the learning content. This is also one powerful feature of xAPI: it's independent of the platform hosting it.

So we made some JavaScript development inside the learning content (which has been made using Storyline authoring tool) to measure the learner activity (as described above), build the according statement and send it to the LRS.

See below two code examples:

Here’s the code to send a statement:

```
function sendStatement(verb, verbId, object, objectId, user_email) {
   var data = JSON.stringify({
      actor: {
         name: "UPTOOL",
         mbox: "mailto:" + user_email,
      },
      verb: {
         id: verbId,
         display: { "en-US": verb },
      },
      object: {
         id: objectId,
         definition: {
            name: { "en-US": object },
         },
      },
   });

   var xhr = new XMLHttpRequest();

   xhr.addEventListener("readystatechange", function () {
      if (this.readyState === 4) {
         console.log(this.responseText);
      }
   });

   xhr.open("POST", "https://lrs.harkhan.com/trax/api/dc802fe8-1e08-4a5b-bf5b-
4fb9b09c28b9/xapi/std/statements");
   xhr.setRequestHeader("Content-Type", "application/json");
   xhr.setRequestHeader("X-Experience-API-Version", "1.0.3");
   xhr.setRequestHeader("Authorization", "Basic " + toBase64("haikara:UQWuQFOZo"));
   xhr.send(data);
}
```

Here’s the code to get a previous statement from the LRS (to update previous data):

```
const statementPromise = new Promise((resolve, reject) => {
      var data = "";
      var xhr = new XMLHttpRequest();
```

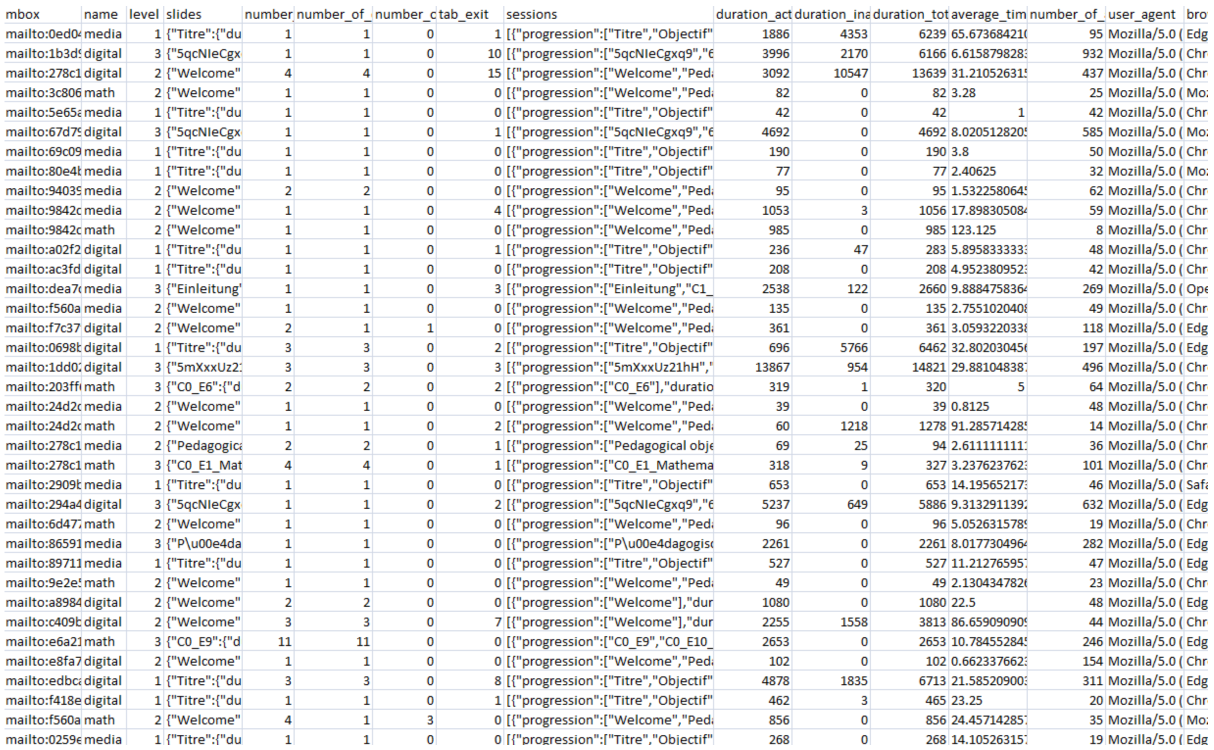
```
        //xhr.withCredentials = true;

        xhr.addEventListener("readystatechange", function () {
            if (this.readyState === 4) {
                if (this.responseText && JSON.parse(this.responseText)) {
                    let r = JSON.parse(this.responseText);
                    stats = r && r.statements && r.statements.length > 0 ?
r.statements[0].object.definition.name["en-US"] : false;
                    if (isJson(stats)) {
                        //localStorage.setItem("uptool_stats", stats);
                        stats = JSON.parse(stats);
                        resolve(stats && Object.keys(stats).length > 0 ? stats : false);
                    } else resolve(false);
                }
            }
        });

        xhr.open("GET", "https://lrs.harkhan.com/trax/api/dc802fe8-1e08-4a5b-bf5b-
4fb9b09c28b9/xapi/std/statements?limit=1&agent={%22mbox%22%3A%20%22mailto%3A"
+ hashUserID + "%40" + window.location.hostname + "%22}&verb=" +
encodeURI(lesson_url) + "");
        xhr.setRequestHeader("X-Experience-API-Version", "1.0.3");
        xhr.setRequestHeader("Authorization", "Basic " + toBase64("haikara:UQWuQFOZo"));
        xhr.send(data);
    });
```

3.3.5. Data produced

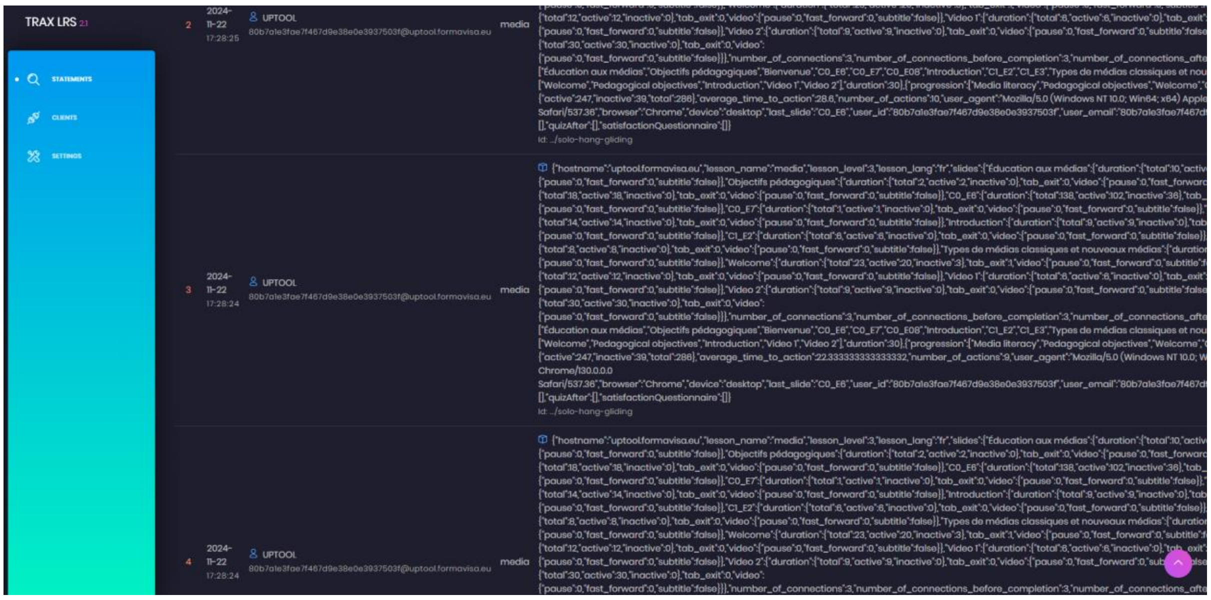
The data stored inside the LRS may be sorted in different ways.
Basically we have one record per user per media per topic (see above how the content has been structured). Here’s an example of a raw extract:



mbox	name	level	slides	number	number_of_	number_c	tab_exit	sessions	duration_act	duration_ina	duration_tot	average_tim	number_of_	user_agent	bro
mailto:0ed0	media	1	{"Titre":{"du	1	1	0	1	[{"progression":["Titre","Objectif"	1886	4353	6239	65.67368421	95	Mozilla/5.0 (	Edg
mailto:1b3d	digital	3	{"5qcNIeCgx	1	1	0	10	[{"progression":["5qcNIeCgxq9","6	3996	2170	6166	6.61587982	932	Mozilla/5.0 (	Chr
mailto:278c1	digital	2	{"Welcome"	4	4	0	15	[{"progression":["Welcome","Peda	3092	10547	13639	31.21052631	437	Mozilla/5.0 (	Chr
mailto:3c806	math	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	82	0	82	3.28	25	Mozilla/5.0 (	Mo
mailto:5e65	media	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	42	0	42	1	42	Mozilla/5.0 (	Chr
mailto:67d7	digital	3	{"5qcNIeCgx	1	1	0	1	[{"progression":["5qcNIeCgxq9","6	4692	0	4692	8.02051282	585	Mozilla/5.0 (	Mo
mailto:69c09	media	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	190	0	190	3.8	50	Mozilla/5.0 (	Chr
mailto:80e4	media	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	77	0	77	2.40625	32	Mozilla/5.0 (	Mo
mailto:9403	media	2	{"Welcome"	2	2	0	0	[{"progression":["Welcome","Peda	95	0	95	1.53225806	62	Mozilla/5.0 (	Chr
mailto:9842	media	2	{"Welcome"	1	1	0	4	[{"progression":["Welcome","Peda	1053	3	1056	17.89830508	59	Mozilla/5.0 (	Chr
mailto:9842	math	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	985	0	985	123.125	8	Mozilla/5.0 (	Chr
mailto:a02f2	digital	1	{"Titre":{"du	1	1	0	1	[{"progression":["Titre","Objectif"	236	47	283	5.89583333	48	Mozilla/5.0 (	Chr
mailto:ac3fd	digital	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	208	0	208	4.95238095	42	Mozilla/5.0 (	Chr
mailto:dea7	media	3	{"Einleitung'	1	1	0	3	[{"progression":["Einleitung","C1_	2538	122	2660	9.88847583	269	Mozilla/5.0 (	Ope
mailto:f560a	media	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	135	0	135	2.75510204	49	Mozilla/5.0 (	Chr
mailto:f7c37	digital	2	{"Welcome"	2	1	1	0	[{"progression":["Welcome","Peda	361	0	361	3.05932203	118	Mozilla/5.0 (	Edg
mailto:0698	digital	1	{"Titre":{"du	3	3	0	2	[{"progression":["Titre","Objectif"	696	5766	6462	32.80203045	197	Mozilla/5.0 (	Edg
mailto:1dd0	digital	3	{"5mXxxUz2	3	3	0	3	[{"progression":["5mXxxUz21hH","	13867	954	14821	29.88104838	496	Mozilla/5.0 (	Chr
mailto:203ff	math	3	{"C0_E6":{"d	2	2	0	2	[{"progression":["C0_E6"],"duratio	319	1	320	5	64	Mozilla/5.0 (	Chr
mailto:24d2	media	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	39	0	39	0.8125	48	Mozilla/5.0 (	Chr
mailto:24d2	math	2	{"Welcome"	1	1	0	2	[{"progression":["Welcome","Peda	60	1218	1278	91.28571428	14	Mozilla/5.0 (	Chr
mailto:278c1	media	2	{"Pedagogica	2	2	0	1	[{"progression":["Pedagogical obje	69	25	94	2.61111111	36	Mozilla/5.0 (	Chr
mailto:278c1	math	3	{"C0_E1_Mat	4	4	0	1	[{"progression":["C0_E1_Mathema	318	9	327	3.23762376	101	Mozilla/5.0 (	Chr
mailto:2909	media	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	653	0	653	14.19565217	46	Mozilla/5.0 (	Safa
mailto:294a4	digital	3	{"5qcNIeCgx	1	1	0	2	[{"progression":["5qcNIeCgxq9","6	5237	649	5886	9.31329113	632	Mozilla/5.0 (	Edg
mailto:6d47	math	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	96	0	96	5.05263157	19	Mozilla/5.0 (	Chr
mailto:86591	media	3	{"P\u00e4da	1	1	0	0	[{"progression":["P\u00e4dagogis	2261	0	2261	8.01773049	282	Mozilla/5.0 (	Edg
mailto:89711	media	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	527	0	527	11.21276595	47	Mozilla/5.0 (	Edg
mailto:9e2e	math	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	49	0	49	2.13043478	23	Mozilla/5.0 (	Chr
mailto:a8984	digital	2	{"Welcome"	2	2	0	0	[{"progression":["Welcome"],"dur	1080	0	1080	22.5	48	Mozilla/5.0 (	Edg
mailto:c409	digital	2	{"Welcome"	3	3	0	7	[{"progression":["Welcome"],"dur	2255	1558	3813	86.65909090	44	Mozilla/5.0 (	Chr
mailto:e6a21	math	3	{"C0_E9":{"d	11	11	0	0	[{"progression":["C0_E9","C0_E10_	2653	0	2653	10.78455284	246	Mozilla/5.0 (	Edg
mailto:e8fa7	digital	2	{"Welcome"	1	1	0	0	[{"progression":["Welcome","Peda	102	0	102	0.66233766	154	Mozilla/5.0 (	Chr
mailto:edbc	digital	1	{"Titre":{"du	3	3	0	8	[{"progression":["Titre","Objectif"	4878	1835	6713	21.58520900	311	Mozilla/5.0 (	Edg
mailto:f418e	digital	1	{"Titre":{"du	1	1	0	1	[{"progression":["Titre","Objectif"	462	3	465	23.25	20	Mozilla/5.0 (	Chr
mailto:f560a	math	2	{"Welcome"	4	1	3	0	[{"progression":["Welcome","Peda	856	0	856	24.45714285	35	Mozilla/5.0 (	Mo
mailto:0259	media	1	{"Titre":{"du	1	1	0	0	[{"progression":["Titre","Objectif"	268	0	268	14.10526315	19	Mozilla/5.0 (	Edg

But of course we can query the database to produce the stats we want.

Here is a view of the statements stored in TRAX LRS:



2 2024-11-22 17:28:25 UPTOOL 80b7a1e3fae7f467d9e38e0e3937503f@uptool.formavisa.eu media

```
{"total":12,"active":12,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Video 1":{"duration":{"total":6,"active":6,"inactive":0},"tab_exit"
{"pause":0,"fast_forward":0,"subtitle":false}},"Video 2":{"duration":{"total":9,"active":9,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false
{"total":30,"active":30,"inactive":0},"tab_exit":0,"video":
{"pause":0,"fast_forward":0,"subtitle":false}}},"number_of_connections":3,"number_of_connections_before_completion":3,"number_of_connections_afte
["Éducation aux médias","Objectifs pédagogiques","Bienvenue","C0_E6","C0_E7","C0_E08","Introduction","C1_E2","C1_E3","Types de médias classiques et nou
["Welcome","Pedagogical objectives","Introduction","Video 1","Video 2"],"duration":30},{"progression":["Media literacy","Pedagogical objectives","Welcome","
{"active":247,"inactive":39,"total":286},"average_time_to_action":28.6,"number_of_actions":10,"user_agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) Apple
Safari/537.36","browser":"Chrome","device":"desktop","last_slide":"C0_E6","user_id":"80b7a1e3fae7f467d9e38e0e3937503f","user_email":"80b7a1e3fae7f467d
[],"quizAfter":[],"satisfactionQuestionnaire":[]}
Id: .../solo-hang-gliding
```

3 2024-11-22 17:28:24 UPTOOL 80b7a1e3fae7f467d9e38e0e3937503f@uptool.formavisa.eu media

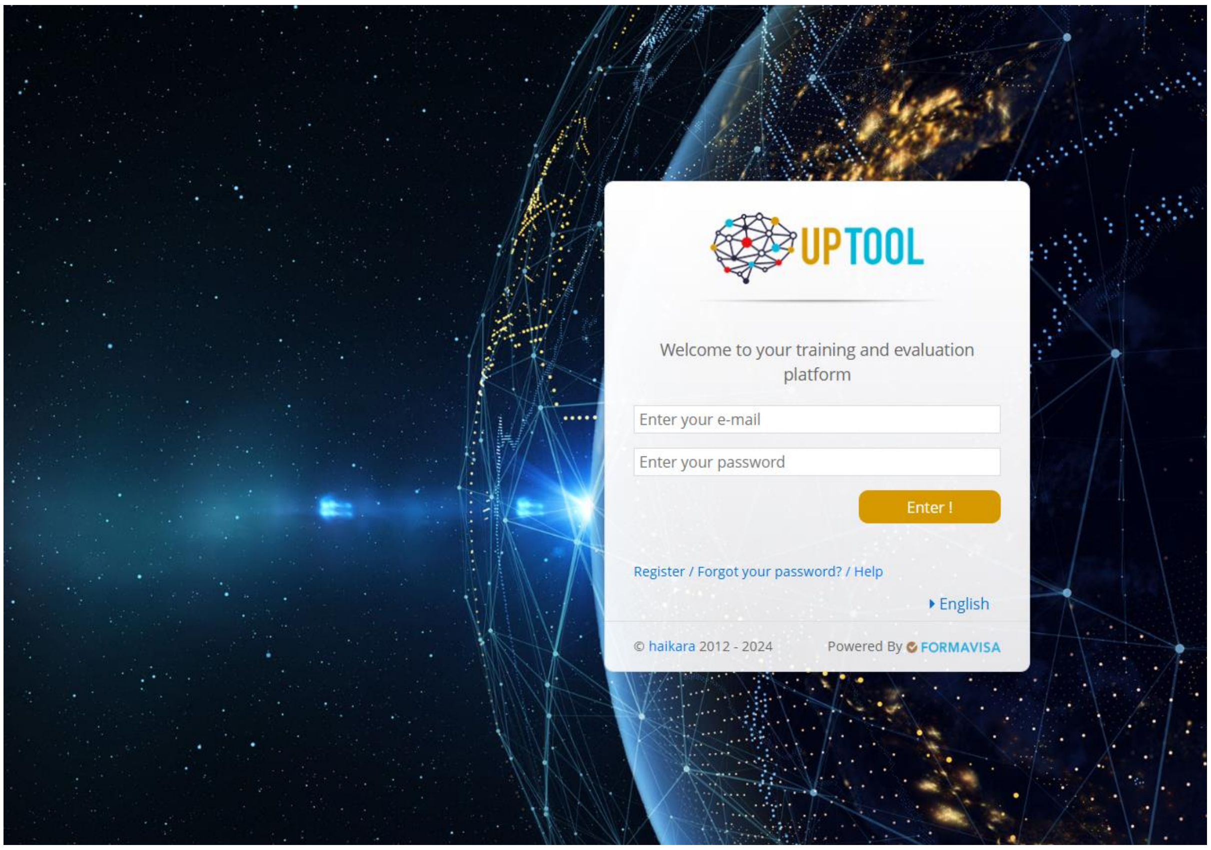
```
{"hostname":"uptool.formavisa.eu","lesson_name":"media","lesson_level":3,"lesson_lang":"fr","slides":{"Éducation aux médias":{"duration":{"total":10,"activ
{"pause":0,"fast_forward":0,"subtitle":false}},"Objectifs pédagogiques":{"duration":{"total":2,"active":2,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forwar
{"total":18,"active":18,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"C0_E6":{"duration":{"total":138,"active":102,"inactive":36},"tab_
{"pause":0,"fast_forward":0,"subtitle":false}},"C0_E7":{"duration":{"total":1,"active":1,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"
{"total":14,"active":14,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Introduction":{"duration":{"total":9,"active":9,"inactive":0},"tab
{"pause":0,"fast_forward":0,"subtitle":false}},"C1_E2":{"duration":{"total":6,"active":6,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}}
{"total":8,"active":8,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Types de médias classiques et nouveaux médias":{"duratio
{"pause":0,"fast_forward":0,"subtitle":false}},"Welcome":{"duration":{"total":23,"active":20,"inactive":3},"tab_exit":1,"video":{"pause":0,"fast_forward":0,"subtitle":f
{"total":12,"active":12,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Video 1":{"duration":{"total":6,"active":6,"inactive":0},"tab_exit"
{"pause":0,"fast_forward":0,"subtitle":false}},"Video 2":{"duration":{"total":9,"active":9,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false
{"total":30,"active":30,"inactive":0},"tab_exit":0,"video":
{"pause":0,"fast_forward":0,"subtitle":false}}},"number_of_connections":3,"number_of_connections_before_completion":3,"number_of_connections_afte
["Éducation aux médias","Objectifs pédagogiques","Bienvenue","C0_E6","C0_E7","C0_E08","Introduction","C1_E2","C1_E3","Types de médias classiques et nou
["Welcome","Pedagogical objectives","Introduction","Video 1","Video 2"],"duration":30},{"progression":["Media literacy","Pedagogical objectives","Welcome","
{"active":247,"inactive":39,"total":286},"average_time_to_action":22.333333333333332,"number_of_actions":9,"user_agent":"Mozilla/5.0 (Windows NT 10.0; W
Chrome/130.0.0.0
Safari/537.36","browser":"Chrome","device":"desktop","last_slide":"C0_E6","user_id":"80b7a1e3fae7f467d9e38e0e3937503f","user_email":"80b7a1e3fae7f467d
[],"quizAfter":[],"satisfactionQuestionnaire":[]}
Id: .../solo-hang-gliding
```

4 2024-11-22 17:28:24 UPTOOL 80b7a1e3fae7f467d9e38e0e3937503f@uptool.formavisa.eu media

```
{"hostname":"uptool.formavisa.eu","lesson_name":"media","lesson_level":3,"lesson_lang":"fr","slides":{"Éducation aux médias":{"duration":{"total":10,"activ
{"pause":0,"fast_forward":0,"subtitle":false}},"Objectifs pédagogiques":{"duration":{"total":2,"active":2,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forwar
{"total":18,"active":18,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"C0_E6":{"duration":{"total":138,"active":102,"inactive":36},"tab_
{"pause":0,"fast_forward":0,"subtitle":false}},"C0_E7":{"duration":{"total":1,"active":1,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"
{"total":14,"active":14,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Introduction":{"duration":{"total":9,"active":9,"inactive":0},"tab
{"pause":0,"fast_forward":0,"subtitle":false}},"C1_E2":{"duration":{"total":6,"active":6,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}}
{"total":8,"active":8,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Types de médias classiques et nouveaux médias":{"duratio
{"pause":0,"fast_forward":0,"subtitle":false}},"Welcome":{"duration":{"total":23,"active":20,"inactive":3},"tab_exit":1,"video":{"pause":0,"fast_forward":0,"subtitle":f
{"total":12,"active":12,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"subtitle":false}},"Video 1":{"duration":{"total":6,"active":6,"inactive":0},"tab_exit"
{"pause":0,"fast_forward":0,"subtitle":false}},"Video 2":{"duration":{"total":9,"active":9,"inactive":0},"tab_exit":0,"video":{"pause":0,"fast_forward":0,"sub
{"total":30,"active":30,"inactive":0},"tab_exit":0,"video":
{"pause":0,"fast_forward":0,"subtitle":false}}},"number_of_connections":3,"number_of_connections_before_completion":3,"number_of_connections_afte
```

5. User access through FORMAVISA

https://uptool.formavisa.eu/


Welcome to your training and evaluation platform

Enter your e-mail

Enter your password

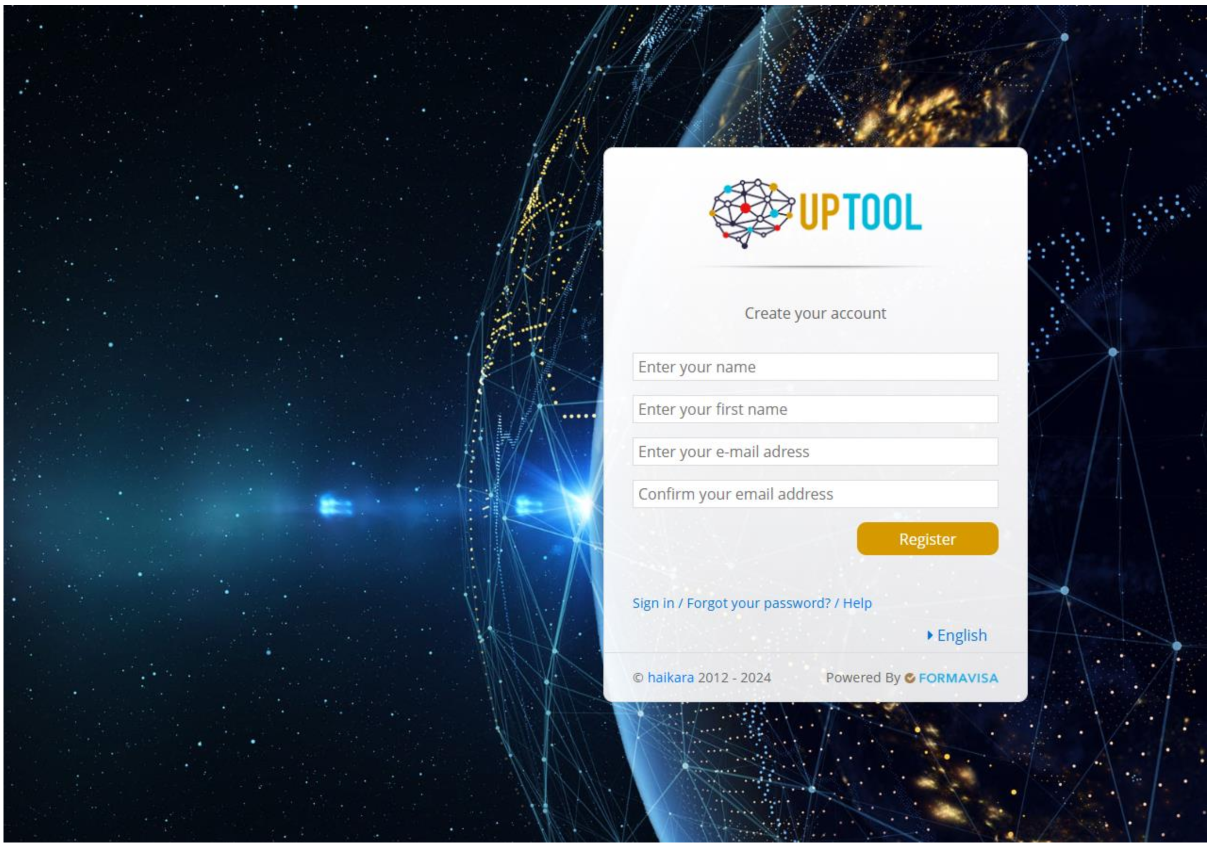
Enter !

Register / Forgot your password? / Help

English

© haikara 2012 - 2024 Powered By FORMAVISA

By clicking on the “Register” link the user can create his account using this form:


Create your account

Enter your name

Enter your first name

Enter your e-mail adress

Confirm your email address

Register

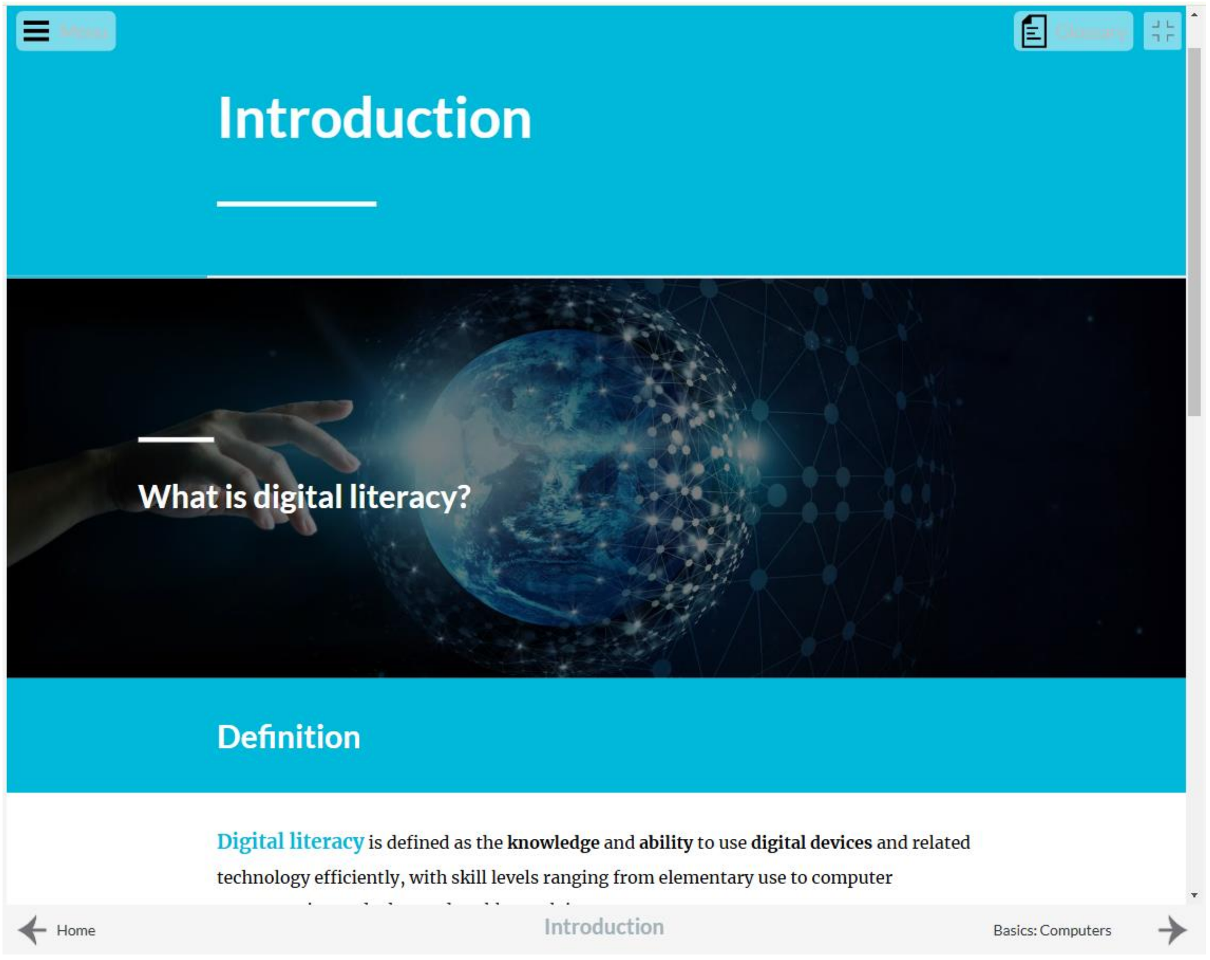
Sign in / Forgot your password? / Help

English

© haikara 2012 - 2024 Powered By FORMAVISA

Once registered the user has access to one lesson of each topic (see above) in a different kind of content (see above).
The first choice of content type is random, and then calculated, to have a distribution of each over the total population.

Example of text-image content on the “Digital” topic:



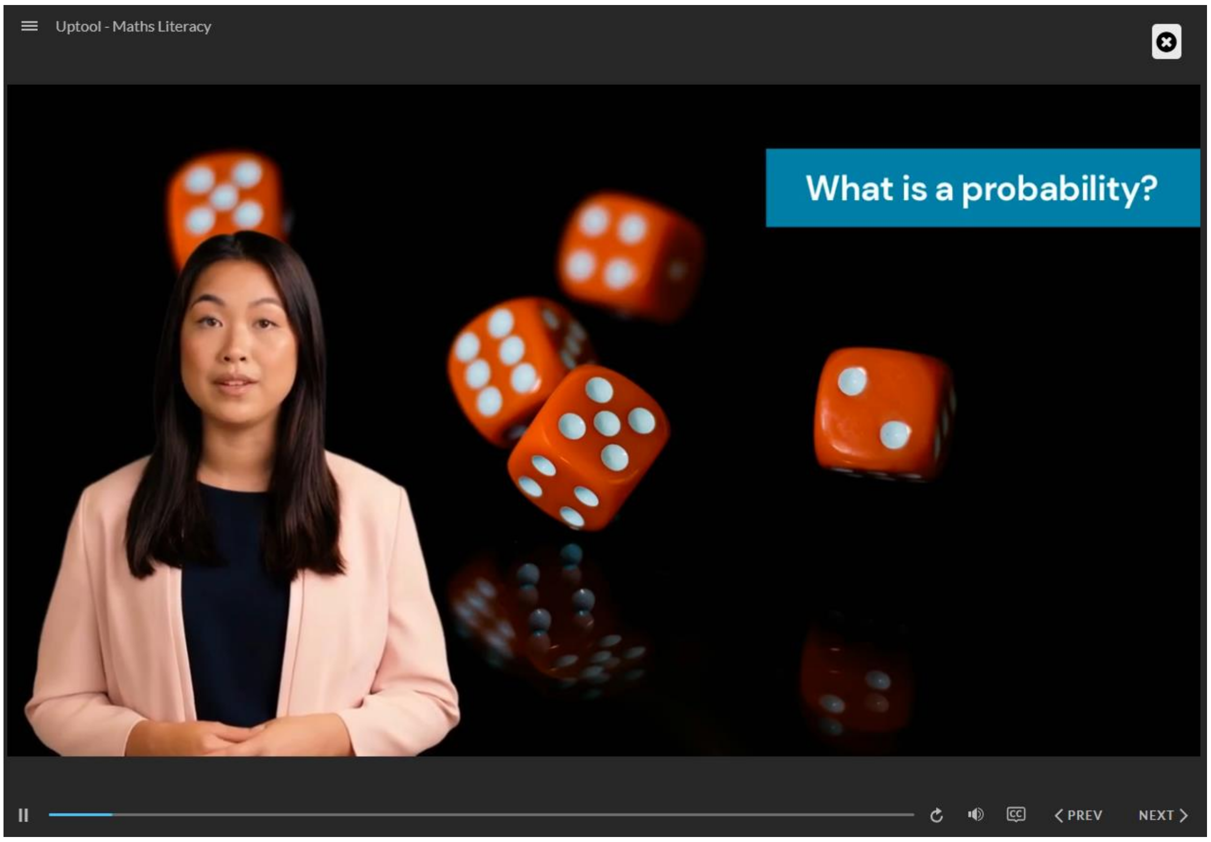
Introduction

What is digital literacy?

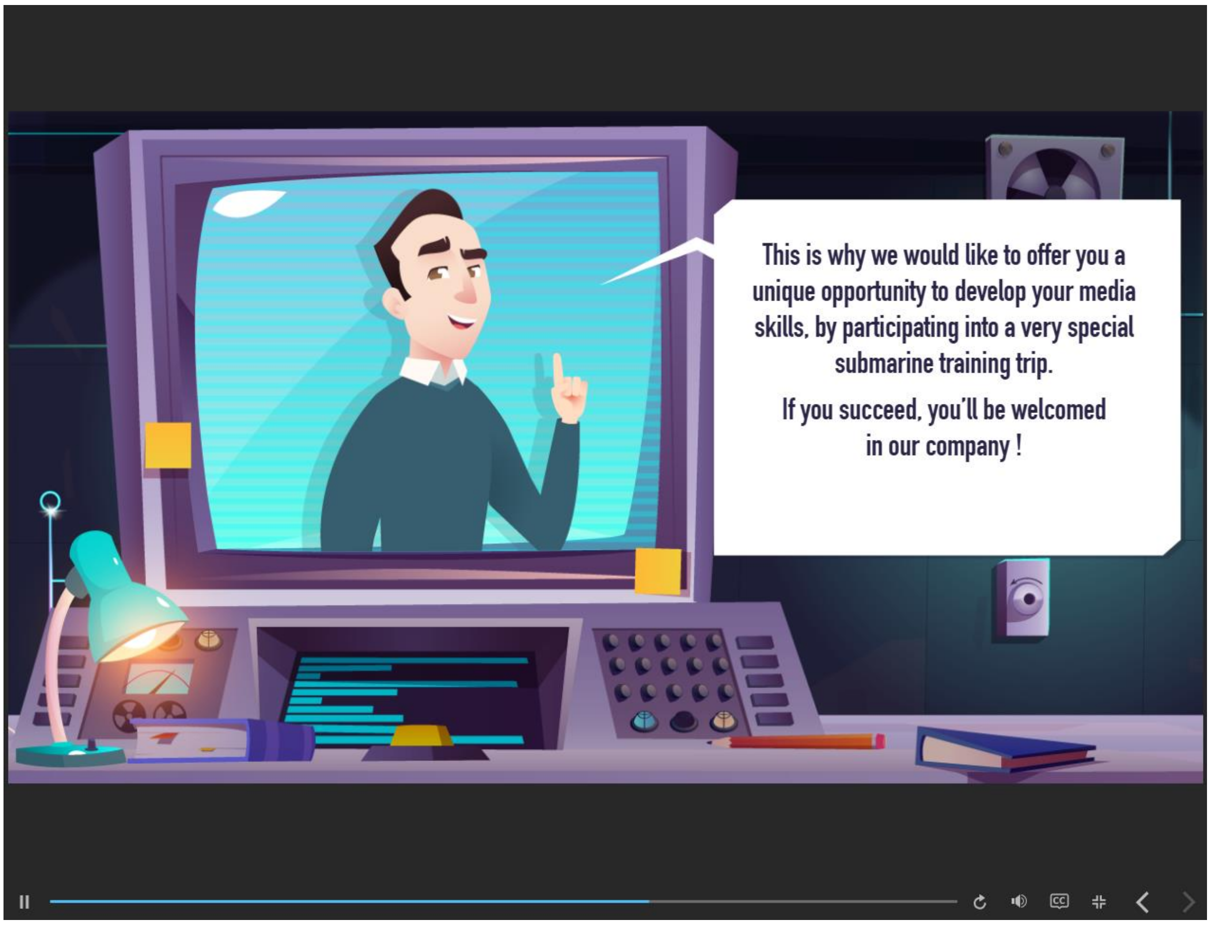
Definition

Digital literacy is defined as the knowledge and ability to use digital devices and related technology efficiently, with skill levels ranging from elementary use to computer

Example of video content on the “Math” topic:



Example of interactive content on the “Media” topic:


This is why we would like to offer you a unique opportunity to develop your media skills, by participating into a very special submarine training trip.

If you succeed, you'll be welcomed in our company !

CONCLUSION

All these developments have led to a modular architecture and a stable environment. It is possible to deploy the same training content on other platforms and to collect user consultation data. This makes the project easily scalable. The variety of data enables all aspects of a learner's behaviour to be measured.
UPTOOL could make it easy to implement a training effectiveness measurement tool in all elearnings.