Functionality: The FSL Hub/Pôle FLS

1. Preamble

The functionality information listed here is provided by the client and is preliminary and in DRAFT format.
Comments from Prof Baljko are prefixed 'MB:' and shown in RED
You will need to read this and gather and interpret the information using your **judgement**.

2. Functionality Description Format

Each of the functions described below employs the following description format template

Functionality	Subsection	Explanation	Notes
1. Needs	1.1 Needs Addressed	What specific needs does this functionality address from all the team volets (groups).	Volets requirements/needs are given to us from a survey we did.
	1.2 Innovation	What is innovative about the chosen type of functionality (e.g. say we choose gallery view for the viewing functionality, then what is innovative is that it's more aesthetically pleasing and personalized by our designer).	
2. Functionality Purpose	2.1 Functionality description of deliverables	More details on how the functionality is expected to work (basic and advanced).	
	2.2 Success criteria for a viable product	What are the points that if met through the design/implementation, then we can say that the functionality deliverable was successful?	Note that these success criteria depend on the different types of functionality descriptions listed in 2.1 above.
3. Examples of Similar Functionalities		Provide as many sample hubs that satisfy as many different types of the functionality (so if there is in search functionality, both basic and advanced, then show all the different types of basic and advanced searches on different hubs).	The sample hubs provided were extracted from our immersive environmental scan.
4. User Paths for this Functionality	None.	How would a user use the functionality if implemented on our hub (note that if there is more than one type of the same functionality, then all different types have their user path).	
5. Functional and Non-functional	5.1 Functional requirements for	What are the functional (technical) requirements for each type of the same	Note that we followed Volere 18 definition of

Requirements	this functionality	functionality (in details).	functional and non-functional requirements.
	5.2 Non-Functional requirements for this functionality	What are the non-functional (non-technical) requirements for each type of the same functionality (in details).	Note that we followed Volere 18 definition of functional and non-functional requirements.

3. Functionality Descriptions

3.1. Personalized Dashboard

MB: sections 4 and 5 are missing for this functionality

1. Needs
1.1 Needs addressed
What user needs are being addressed by this functionality? If specific to certain stakeholders, please indicate who.
Dashboarding allows users to personalize their experience on the hub in different ways, such as: store certain content of interest to them, view their own activity on the hub, and be able to view a library of their own uploaded/created content and/or content they saved from the hub itself.
Innovation
What makes this functionality innovative? If specific to certain stakeholders, please indicate who.
Every user can personalize their own experience on the hub by changing their own dashboard in a format they desire.
2. Functionality purpose
2.1 Functionality description of deliverables
What are the deliverables for this functionality? E.g. a forum, with threads
Volet 1b: 15.1 - MAYBE: Simple design but some of the metrics should have the option of being hidden. Could be combined with 15.3 15.2 - NO: UI doesn't appear to be as intuitive as 15.1 or 15.3 15.3 - YES: An activity overview is important for tracking student engagement and should be prioritized (perhaps in tandem with 15.1) I think Idéllo's setup is interesting. **Volet 2:** We like 15.2. It would be good to incorporate some of 15.1 to be able to track the interactions with our posts such as how many views and downloads were completed. **Volet 3:** None at the moment. **Volet 4:** 15.1, 15.2, 15.3 would be useful for the teachers of the courses **(I believe they are refering to the functionalities document:** https://docs.google.com/document/d/1uIR8-jNT01F_4dChfurUyZweKn6xzApLfaQf8Cne4Is/edit?usp=sharing),.

2.2 Success criteria for viable product

This section is to explain what the success criteria for this functionality is (UX, UI, development, engagement).

- Users are able to view the content they saved in an organized manner and be able to retreive the content easily.

- Users are able to view their own content they uploaded or created on the hub itself easily and be able to retrieve the content in them easily.

- Users are able to view their activity on the hub and what they iteracted with so far.

3. Examples of similar functionalities

Provide a few examples of sites with similar functionalities, and highlight the features that make them strong.

ID.	Feature	Site and functionality highlights
3.0	Library	Users can view the content they saved on the hub and the content they created/uploaded in an organized library format [link]
3.1	Activity Level	You can view your own activity level and profile/dashboard in a nice and intuitive manner [link].
3.2	Activity overview	Here teachers can track essential metrics about the classroom they are teaching and how students are engaging with the content. This dashboard is innovative and makes the platform more user friendly for teachers.[link]
3.3	Personalization	Personalized Dashboard on Idello. [link]

4. User Paths for this functionality

List all the possible user paths for this functionality here.

ID.	Priority (high, low)	User paths
4.0		Example: A teacher opens the website and sees the home page. They are an advanced user and go to the advanced search option. They wish to find grammar content and type in grammar getting search results for their topic. They can also select a number of other variables within the advance search options such as - type, language, proficiency level etc.
4.1		
4.2		
4.3		
4.4		
4.5		

5. Functional and Non-Functional Requirements

List all the functional and non-functional requirements here.

5.1 Functional Requirements for this functionality

ID.	Priority	Requirements
5.1.0		Example: You should be able to log in to the forum to see the posts.
5.1.1		
5.1.2		
5.1.3		

5.1.4		
5.1.5		
5.2 Non-Functional Requirements for this functionality		
ID.	Priority	Requirements
5.2.0		Example: the forum threads should load completely within 20 seconds
5.2.1		
5.2.2		
5.2.3		
5.2.4		
5.2.5		

3.2. Download

1. Needs
1.1 Needs addressed
What user needs are being addressed by this functionality? If specific to certain stakeholders, please indicate who.
The download functionality allows users to retain a copy of resources both on their local drive and on their online repository.
Innovation
What makes this functionality innovative? If specific to certain stakeholders, please indicate who.
Download functionality in the context of the FSL repository is innovative as it allows users to choose between what storage location they prefer to keep their resources in. Online storage has become very inexpensive nowadays with cloud services like AWS, Azure and GCP offering simple storage services for cents/month. Providing users with this option is innovative because they can easily access all their resources from one place and optionally upload them into the same storage space. This will help reduce any barriers for uploading new content on our platform
2. Functionality purpose
2.1 Functionality description of deliverables
What are the deliverables for this functionality? E.g. a forum, with threads
The purpose of the download functionality is to provide two options of storing a copy of a resource found on the accelerator. Specific deliverables for this functionality include: - Ability to archive a particular resource to a user's sandbox/**personal storage repository** - Ability to download resource in a **non proprietary data format** (e.g .pdf, .csv, .xml, .jpg, .png, .json, .mp3, .mp4, .txt)
2.2 Success criteria for viable product
This section is to explain what the success criteria for this functionality is (UX, UI, development, engagement).
- Users can add (or download) **any resource** from the FSL Accelerator to their personal storage repository OR to their local drive - Users can store, organize and find specific resources they have downloaded in the past in one place - Users can sort rsources based on file type, date added, and other criteria

3. Examples of similar functionalities		
Provide a few examples of sites with similar functionalities, and highlight the features that make them strong.		
ID.	Feature	Site and functionality highlights
3.0		Example: [hyperlink] and highlights
3.1	Fork Downloads	Similar to the concept of "Forking" on github. Users can download a resource from another user's repository and have a copy of it. They can edit and repurpose however they like. Additionally the user can pull any updates from original resource owner View here [Github Fork Feature] https://github.com/
3.2	Open downloads	You do not need to create a SkillsCommons account to download materials.
3.3	Exporting	Download but possibly in different formats.
3.4	Cloud Download	saving data to virtual servers instead of local drive.
3.5	Autherization to download	Access needed to download some or all content on the hub.
3.6	Bulk Downloading	downloading a full list of resources/a complete database as a zip file.
4. User Paths for this functionality		
List all the possible user paths for this functionality here.		
ID.	Priority (high, low)	User paths
4.0		Example: A teacher opens the website and sees the home page. They are an advanced user and go to the advanced search option. They wish to find grammar content and type in grammar getting search results for their topic. They can also select a number of other variables within the advance search options such as - type, language, proficiency level etc.
4.1		A user searches for a resource on the FSL Platform. Upon clicking on a specific resource, they are able to view Download options for the specific resource.
4.2		If the user is not logged in, they are only able to download the resource to their local drive
4.3		If the user is logged in, they are able to download the resource and save it to their online repository.
4.4		
4.5		
5. Functional and Non-Functional Requirements		
List all the functional and non-functional requirements here.		
5.1 Functional Requirements for this functionality		
ID.	Priority	Requirements
5.1.0		Example: You should be able to log in to the forum to see the posts.
		The system shall allow individuals with personal user accounts to save multiple “My Downloads” lists of resource objects and searches
5.1.1		The system shall allow a user to click on a download link associated with a resource object and initiate a download process to the users online repo or local drive
5.1.2		The system shall allow user to choose their preferred file format for the download
5.1.3		The system shall generate (as part of the personal online download repo), a persistent URL that links to the version of the resource object being downloaded

5.1.4		The system shall allow the user to select a view of their downloaded resources (for example, lesson plans, study guides, recent downloads)
5.1.5		The system shall provide ability to link from a personal storage repository download back to where the content was originally saved
		The system shall provide contextual help to users who do not know how the download functionality works
5.2 Non-Functional Requirements for this functionality		
ID.	Priority	Requirements
5.2.0		Example: the forum threads should load completely within 20 seconds
5.2.1		The resource should have an appropriate license for download
5.2.2		
5.2.3		
5.2.4		
5.2.5		

3.3. Upload

1. Needs
1.1 Needs addressed
What user needs are being addressed by this functionality? If specific to certain stakeholders, please indicate who.
The upload functionality allows users to post a copy of their resources to our online repository or their specific dashboard.
Innovation
What makes this functionality innovative? If specific to certain stakeholders, please indicate who.
A hybrid model of the upload functionality that allows users to upload in open formats as well as proprietary formats is an innovative functionality for the accelerator.
2. Functionality purpose
2.1 Functionality description of deliverables
What are the deliverables for this functionality? E.g. a forum, with threads
The purpose of this functionality is to allow users to select a resource from their local drive (or potentially import from other sources) and upload it to our database. This resource should be traceable to the uploader's account and should be searchable by other users unless specifically specified in the upload process. - Working upload functionality that posts resources to our Database - Working upload form to capture details about the rsource being uploaded (e.g Resource type, License, thumbnail, data format etc) - A moderator panel for approving uploads

2.2 Success criteria for viable product

This section is to explain what the success criteria for this functionality is (UX, UI, development, engagement).

The upload functionality should be able to capture and parse multiple data formats chosen for our accelerator (hybrid of open and proprietary ones, .pdf, .csv, .xlms, .docx, .pages, .rtf , .jpg, .png, .mp3 .mp4 .wav)

The upload functionality should be available for multiple data and media types including resource imports from other platforms (e.g Google classrooms)

3. Examples of similar functionalities

Provide a few examples of sites with similar functionalities, and highlight the features that make them strong.

ID.	Feature	Site and functionality highlights
3.0		Example: [hyperlink] and highlights
3.1	Material type	The SkillsCommons Repository houses two types of materials, Learning Resource Materials and Program Support Materials To contribute/upload into one of these repositories you need a Skillscommons account. See more here [SkillsCommons - Upload Material] https://support.skillscommons.org/home/contribute-manage/contribute-materials/ [SkillsCommons - Contribute and Manage material] https://support.skillscommons.org/home/contribute-manage/
3.2	Google Uploads	On Google drive, you can upload files into private or shared folders by simply drag and dropping files into the workspace. See more here [Google upload - files and folders] https://support.google.com/drive/answer/2424368?co=GENIE.Platform%3DDesktop&hl=en
3.3	Cloud Uploads	Cloud storage involves stashing data on hardware in a remote physical location, which can be accessed from any device via the interne (clients uploading to the cloud means data is stored in serves which can be accessed by anyone and anywhere).
3.4	Bulk Upload	Uploading a full list of resources/a complete database.
3.5	Requirements for the upload	Content must satisfy certain requirements to be uploaded.
3.6	Authorization to upload	Access required to upload content on the hub.
3.7	Upload UIs	Screenshots of upload functionalities that are simply concepts or require user login. View here [2. Upload Features and UI] https://medium.com/@paulcharlesowe/deep-dive-functionalities-ca85e6049436
3.8	Modalities	Content modality options. View here [2. Upload Features and UI] https://medium.com/@paulcharlesowe/deep-dive-functionalities-ca85e6049436
3.9	Drag and drop and file browser	Drag and drop and/or file browsing functionality for uploading content. View here [2. Upload Features and UI] https://medium.com/@paulcharlesowe/deep-dive-functionalities-ca85e6049436
3.11	Upload detail editor	Concept design of upload metadata tag editor. View here [2. Upload Features and UI] https://medium.com/@paulcharlesowe/deep-dive-functionalities-ca85e6049436
3.12	File type recognition	Concept design of file type recognition after the user uploads a file

		View here [2. Upload Features and UI]
4. User Paths for this functionality		
List all the possible user paths for this functionality here.		
ID.	Priority (high, low)	User paths
4.0		Example: A teacher opens the website and sees the home page. They are an advanced user and go to the advanced search option. They wish to find grammar content and type in grammar getting search results for their topic. They can also select a number of other variables within the advance search options such as - type, language, proficiency level etc.
4.1		A user begins by clicking "Upload a resource" and is presented with a modal form that captures the basic details about the resource they are about to upload - whether it is from local drive or an import. Other details that may be captured could involve whether the resource has an open data format or is proprietary. This information will allow us to understand how to parse and store the user's upload to our database
4.2		Once a resource is selected and the form is filled, the upload button becomes active and the user can press it to begin uploading.
4.3		
4.4		
4.5		
5. Functional and Non-Functional Requirements		
List all the functional and non-functional requirements here.		
5.1 Functional Requirements for this functionality		
ID.	Priority	Requirements
5.1.0		Example: You should be able to log in to the forum to see the posts.
5.1.1		The upload functionality should be able to parse and store open and proprietary data and media formats
5.1.2		The upload functionality should be able to integrate with other specified LMS such as google classrooms to allow for easy imports into our database
5.1.3		The upload should be traceable back to the uploader
5.1.4		Only users with accounts and ones that are already signed in may access upload functionality
5.1.5		Recently uploaded content should be accepted into our system but should be subject to moderator approval before appearing to the general public - especially for media content that cannot be efficiently parsed
5.1.6		A moderator panel should exist, where moderators can monitor, approve/disapprove recently uploaded content
5.1.7		Only content approved by moderator should be searchable by all users
5.1.8		Allow for full customization of licensing option
5.2 Non-Functional Requirements for this functionality		
ID.	Priority	Requirements
5.2.0		Example: the forum threads should load completely within 20 seconds
5.2.1		Uploaded content should adhere to all licensing guidelines
5.2.2		During the upload, the screen shows a progress bar
5.2.3		Once upload is done, a thumbnail of the image is displayed and options are presented for title of the file, description in any language, and

		category(ies).
5.2.4		
5.2.5		

3.4. Initial Gallery View

1. Needs		
1.1 Needs addressed		
The hub has a lot of resources - we need a way to view all the content on the hub in an organized and non-overwhelming way. This addresses the needs of all stakeholders. All the stakeholders agree on gallery view and not an interactive map nor a list view.		
Innovation		
What makes this functionality innovative? If specific to certain stakeholders, please indicate who.		
A gallery view is both aesthetically pleasing and personalized for our hub by our designer.		
2. Functionality purpose		
2.1 Functionality description of deliverables		
1. Deliverable: a gallery view of all the content on the hub with a representative picture for each specific content. In other words, the pictures in the gallery must give an indication of what is in the link when pressing the picture. 2. Deliverable: the gallery must be organized by categories representing the tags from the metadata sheet. 3. Deliverable: each picture for each content must have a brief description of the content in it (as shown below in Airtable link). 4. Deliverable: each category of pictures must have a clear tag on top showing what the content below belong to (e.g. ck-12 division based on type of content/courses).		
2.2 Success criteria for viable product		
This section is to explain what the success criteria for this functionality is (UX, UI, development, engagement).		
- Non-overwhelming gallery view of all the content on the hub.		
- All the content of the hub are represented by a unique pictures.		
- Organized content based on tags from the matadata sheet in an easy to understand format.		
3. Examples of Gallery View		
Provide a few examples of sites with similar functionalities, and highlight the features that make them strong.		
ID.	Feature	Site and functionality highlights
3.0		Example: [hyperlink] and highlights
3.1.1	Scrollable carousel	Coursera implements multiple carousel components that are easily scrollable and make it seem like a gallery of resources. Each carousel is spearated by a main theme such as - Most Popular courses, or Top rated courses in Language learning, etc View it here [Coursera - Scrollable carousel] https://www.coursera.org/browse/language-learning
3.1.2	Gallery view organized by subject	Ck12 has a gallery view for its resources and organizes it by subject. Each resource has a picture associated with it View it here [link]. https://www.ck12.org/student/

3.1.3	Gallery view	Idello [link]. https://www.idello.org/en/explore-resources
3.1.4	Rounded edges gallery view	Pinterest has a visually appealing interface. They have their gallery items well spaced and the rounded edges make it look great. View it here [Pinterest gallery view] https://www.pinterest.ca/ideas/
3.1.5	gallery view with quick access to ratings	Imagine being able to immediately view ratings/likes on a resource before opening it. Dribbble implements this concept neatly on their site neatly. It allows you to browse resources in a gallery view and find highly rated ones all on one page. View it here [Dribbble ratings on gallery view] https://dribbble.com/tags/image_gallery
3.1.6	Guideline	Airtable [link2] (this is not a hub, nor an actual website with a gallery view. It's a guideline on making a gallery view, but the examples illustrated there are very well created). https://support.airtable.com/hc/en-us/articles/229848887-Guide-to-kanban-view
3. Examples of List View		
	List view	On Twilio users are able to view their resources in list format. For example, after entering a search query here is a demonstration of how list view could look on our platform Try it here [https://support.twilio.com/hc/en-us/search#stq=api&stp=1]
	List view variation with advanced search option	On github, resources (repositories) are returned back in a list view format. What is interesting about it is that it also adds an advanced search feature on the left side whihc conitnues to give users to further refine the current repositories being shown. This view is also effective for resources that are user uploaded. This is because with gallery view we may need to display an image, and we do not want empty images in our search results for users who choose not to upload an image with their resources Try github's list view here [List view for OER Repositories on github] https://github.com/search?q=oer&type=Repositories
	List view with image	Yelp offers a list view for their listings. Here is an example list view for a query searching for French teachers Try it here [Yelp French Teachers - List View] https://github.com/search?q=oer&type=Repositories
4. User Paths for this functionality		
List all the possible user paths for this functionality here.		
ID.	Priority (high, low)	User paths
4.0		Example: A teacher opens the website and sees the home page. They are an advanced user and go to the advanced search option. They wish to find grammar content and type in grammar getting search results for their topic. They can also select a number of other variables within the advance search options such as - type, language, proficiency level etc.
4.1		A user opens the website and sees the home page. In the home page, they are able to view the content on teh hub categorized by tags in the matadata analysis sheet. Once they choose a certain category, then the website will take them to a more specified page with another gallery view based on subtags from the tags in the matadata analysis sheet. E.g. home page shows grades (gr1, gr2, etc.), when choosing gr10 for example, the teacher will see sub-gallery view with content divided by courses (physics, bio, english, etc.).
4.2		
4.3		
4.4		
4.5		

5. Functional and Non-Functional Requirements		
List all the functional and non-functional requirements here.		
5.1 Functional Requirements for this functionality		
ID.	Priority	Requirements
5.1.0		Example: You should be able to log in to the forum to see the posts.
		All site components should be resizable for all screen types (e.g tablets, desktop, laptop, mobile)
5.1.1		Use of highly accessible color palette, fonts, and user-interface design elements.
5.1.2		Text alternatives/ transcriptions for video and audio
5.1.3		Text shall employ adequate contrast and size for readability
5.1.4		All audio and video shall display with player controls (play, pause, volume, quality)
5.1.5		All audio and video should only overlay on the page and users should be able to click away instead of opening a new tab or redirecting to a new page
5.2 Non-Functional Requirements for this functionality		
ID.	Priority	Requirements
5.2.0		Example: the forum threads should load completely within 20 seconds
5.2.1		All views/pages must load and render quickly for the user.
5.2.2		Design for continuous availability of interactive experience.
5.2.3		Follow a consistent style across all pages.
5.2.4		Should follow principles related to the creation of simple, intuitive and easy to use UI/UX (e.g Golden ratio)
5.2.5		All external visual assets shall appear with appropriate credits, sources

3.5. Faceted Navigation

MB: clients are not using the term 'faceted navigation' in preference to 'advanced search'. Note that 'search' generally refers to a context in which the user is presented with nothing (blank) and the blank slate is added to via criteria that match/don't match the search terms. In filter, we start with the full set of items, and (successively) remove items based on criteria (remove items that don't match specified filter terms). Faceted Navigation is an advanced form of structured filtering. See Nelson Norman Group's article "Filters vs. Facets: Definitions" https://www.nngroup.com/articles/filters-vs-facets/
Your task is to interpret the client information below using your own judgment.

1. Needs
1.1 Needs addressed
What user needs are being addressed by this functionality? If specific to certain stakeholders, please indicate who.
The hub has a lot of resources - we need a way to navigate and search for resources that is simple to use. This addresses the needs of all stakeholders. Other needs covered by search functionality includes the ability to specify the exact resources you are looking for.
Innovation

What makes this functionality innovative? If specific to certain stakeholders, please indicate who.

As shown in the link below, it's a different/innovative way of building up to an advanced personalized search.
link https://docs.google.com/document/d/1UzmvKf3JuFRcmO7VK9bHrWe51oyMVLu6k5diBfb6wMw/edit?usp=sharing

2. Functionality purpose

2.1 Functionality description of deliverables

What are the deliverables for this functionality? E.g. a forum, with threads

1. Deliverable: Basic search and advanced (based on tags).
2. Deliverable: search functionality with optional filtering attributes that allow the user to specify their query.
3. Description: Search engines are a category of **information retrieval systems** designed to supply a subset of items from a population based on a specified set of criteria. The structure and function of search engines are largely determined by:
- The nature of the information that is the target of a search;
- The methods used to distinguish a subset of information from a larger population and
- The interface used to conduct search queries and to display query results

2.2 Success criteria for viable product

This section is to explain what the success criteria for this functionality is (UX, UI, development, engagement).

- Advanced search is precise and returns results specifc to the query defined. Success of this functionality would depend on the underlying tagging system used on our content

- Fast queries

3. Examples of similar functionalities

Provide a few examples of sites with similar functionalities, and highlight the features that make them strong.

ID.	Feature	Site and functionality highlights
3.0		Example: [hyperlink] and highlights
3.1.1	Autocomplete	Twilio has a full screen autocomplete interface which filters out content as you type your query Try it here [Twilio Support - Autocomplete] https://support.twilio.com/hc/en-us
3.1.2	"Omni-search", Autocomplete and UI	Elastic has a search bar that is present on any page you visit on their site. Implementation of search on this site is also interesting because it appears from the top of the site making it convenient for users to continue browsing the rest of the site Try it here [Elastic - Omnisearch] https://www.elastic.co/
3.1.3	Toggle between Use cases	CC Search has a unique feature that allows you to specify your use-case before you use their basic search Try it here [CC Basic Search] https://www.elastic.co/
3.1.4	How-to search, UI interface	Ck12 provides you with a short video on how to effectively use their search. The video is accessible as an overlay on the page which makes it easy for users to watch without being redirected elsewhere Try it here [CK12 Basic Search with quick How-to-video] https://www.ck12.org/student/
3.1.5	Autocomplete, Dynamic result set	Swiftype has a search bar that dynamically displays result sets as you type your query. It also has a visually appealing interface

		Try it here Swiftype dynamic search] https://www.ck12.org/student/
3.1.6	Dynamic search UI	Design+code has a generally appealing user interface. Its basic search functionality filters courses dynamically and displays them in a gallery veiw as you type your query Try it here [Design+Code - Dynamic search UI] https://designcode.io/courses
3.1.7	Other ideas for basic search	Multiple basic search examples [Basic Search working compilation] https://designcode.io/courses
3.2 Examples of Advanced Search Options		
	Advanced search, Personalized search, Toggle between DBs	York University's advanced search allows you to toggle between specific databases (e.g Course reserves, Omni libraries, Libraries) It also allows you to personalize search results based on your user account or specific topic interests. It's UI is not that great and may come off as unintuitive for non experienced users. Try it here [York University Advanced Search][1]
3.2.1	Innovative, Intelligent Search	"Amazon Kendra is an intelligent search service powered by machine learning. Kendra reimagines enterprise search for your websites and applications so [users] can easily find the content they are looking for, even when it's scattered across multiple locations and content repositories within your organization. Using Amazon Kendra, you can stop searching through troves of unstructured data and discover the right answers to your questions, when you need them. Amazon Kendra is a fully managed service, so there are no servers to provision, and no machine learning models to build, train, or deploy." - Amazon With Kendra we can use a feature that allows our users to feel more engaged on our site by allowing them to query/voice search our web application and ask for information like "When is your next Stakeholder consultation" and receive answers. It basically allows information from your internal organization to be searchable throught the power of NLP. This may allow us to engage and connect with our users better because we can simulate our organization's presence. Additionally, it is only a service that can be plugged in, instead of bottom up implementation. [Amazon Kendra Features] https://aws.amazon.com/kendra/features/
3.2.2	Advanced search	"Narrow down search results for complex searches by using the Advanced Search page. For example, you can find sites updated in the last 24 hours or images that are in black and white." - Google Google's advanced search is not visually appealing however it has an impressive set of filtering options that i have not seen in other ordinary advanced search features. I think it can serve as an example to get ideas of how a great search engine does it. Try it here [Google Advanced Search - For Websites] https://www.google.com/advanced_search [Google Advance Search - For images] https://www.google.com/advanced_image_search
3.2.3	Refined search techniques	On google, there's a difference between advanced search and refined search. with refined search, users are able to use symbols or words in your search to make your search results more precise. Common search techniques include hashtag search, exclusion of certain words, finding exact matches, and combining searches

[1] https://ocul-yor.primo.exlibrisgroup.com/discovery/search?vid=01OCUL_YOR:YOR_DEFAULT&tab=Everything&search_scope=MyInst_and_CI&mode=advanced&displayMode=full&bulkSize=10&highlight=true&dum=true&query=any,contains,&displayField=all&pcAvailabiltyMode=true&lang=en

		Try it here [Google Refined Search] https://support.google.com/websearch/answer/2466433?hl=en&ref_topic=3081620
3.2.4	Simple advanced search, Overlay advanced search	Twitter's advanced search is interesting because of the mix of simplicity and adavnced search precision it provides users. Its user interface for the advanced search is also great because it displays a modal that is an overlay of the current screen the user is on. Try it here [Twitter Advanced Search] https://twitter.com/search-advanced?lang=en
3.2.5	Advanced search	Idello is worth exploring because it is the closest hub to our accelerator. Their advanced search functionality can serve as our baseline in terms of its filtering options. Idello [link] https://www.idello.org/en/explore-resources?page=1&sort=popularity&layout=grid&query=&categories=
3.2.6	Advanced search layout variation	Crunchbase demonstrates how we can implement a "different" / new layout for advanced search. It may be unintuitive for certain users but its worth exploring it as an option. Crunchbase [link] https://www.crunchbase.com/search/organization.companies/99ea5d44b07789e084984c64229992d8
3.2.7	Advanced search layout variation	Github [GitHub advanced search] https://github.com/search/advanced
3.2.8	Simple Advanced Search	Red Hat is the world's largest open source software company. It has a standard layout for the advanced search feature found on most websites that is easy and well known by users of all skill level. However, It is still able to compartmentalize a lot of advanced search filters into a fairly small number of major categories. We may opt to explore this approach for the accelerator as well instead of having a fancy (innovative, or different) but complicated search functionality. Try it here [Redhat Advanced Search] https://github.com/search/advanced
3.2.9	Relevant search tags for the accelerator	Explore 60+ tags used in other OER hubs' advanced search functionality [link] https://docs.google.com/spreadsheets/d/1SwH4w1yXqjX1fHcYc3IarUmYCMVVQncWcCfW73iHqpk/edit#gid=1511192506

4. User Paths for this functionality		
List all the possible user paths for this functionality here.		
ID.	Priority (high, low)	User paths
4.0		Example: A teacher opens the website and sees the home page. They are an advanced user and go to the advanced search option. They wish to find grammar content and type in grammar getting search results for their topic. They can also select a number of other variables within the advance search options such as - type, language, proficiency level etc.
4.1		A teacher opens the website and sees the home page. They are an advanced user and go to the advanced search option. They wish to find grammar content and type in grammar getting search results for their topic. They can also select a number of other variables within the advance search options such as - type, language, proficiency level etc. Same reasoning and navigation for different stakeholders such as (i) a student, (ii) a parent and (iii) an administrator in a school board.
4.2		Users can progressively build queries (for different entities, such as Hubs, Schools, Events, etc) and configure custom filters. e.g. User starts by choosing school then you get a sub filter (Private/Public) within a sub filter there is a possibility of other subfilters (area, name of school).
4.3		
4.4		

4.5		

5. Functional and Non-Functional Requirements

List all the functional and non-functional requirements here.

5.1 Functional Requirements for this functionality

ID.	Priority	Requirements
5.1.0		Example: You should be able to log in to the forum to see the posts.
5.1.1		Should be able to search through resources hosted on the site by title, or tags where appropriate.
5.1.2		Search should be accessible from every page in the app
5.1.3		Accepts and parse string, numeric, date, and other specified formats on the search interface
5.1.4		Can store search queries from our users into a database for future analysis and optimization
5.1.5		Users should be able to toggle between the underlying source databases they want to search from
5.1.6		Should be able to search **resource tags** and **resource titles** at the same time. This will allow us to return items of different types on the same page at the same time (e.g videos -tagged, articles - titles)
5.1.7		Specifying OR/AND/NOT Search
5.1.8		Should be able to list the query result in ascending alphabetical order; most frequently accessed order, or as specified in our document to the software company
5.1.9		Users should be able to toggle between basic and advanced search

5.2 Non-Functional Requirements for this functionality

ID.	Priority	Requirements
5.2.0		Example: the forum threads should load completely within 20 seconds
5.2.1		Fast queries
5.2.2		Must have a clear metadata structure or diagram for how the system is tagging resources
5.2.3		User friendly, responsive, adaptable
5.2.4		Depending on access control guidelines, we may need to configure our search functionality to provide users with content they are authorized to access only
5.2.5		