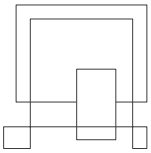


BSc Thesis in Software Technology Engineering

Project Report

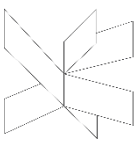
Building a website to promote the CERN Academic Training lectures

Franciska-Leonóra Török, 293171 IT

Supervisors:

Kasper Knop Rasmussen
VIA University College

Maria Dimou
CERN IT, Academic Training



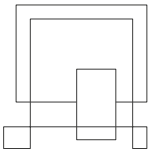
VIA University College

40,679 characters

Software Technology Engineering

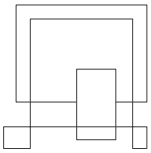
7th Semester

2023



Abbreviations – Acronyms

CERN	European Organization for Nuclear Research
IT	Information Technology (CERN Department)
RCS	Research and Computing Sector (CERN Department)
CDA	Communications, Devices & Applications (CERN Group)
SIS	Scientific Information Service (CERN Group)
ATC	Academic Training Committee
CDS	CERN Document Server
LHC	Large Hadron Collider
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
API	Application Programming Interface
XML	Extensible Markup Language
JSON	JavaScript Object Notation
DBoD	Database On Demand
UI	User Interface
UX	User Experience
SPA	Single Page Application
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
VCRI	Verification and Cross Reference Index



Acknowledgements

I wish to express my gratitude to my supervisor at CERN, Maria Dimou, for her continuous support and motivation throughout my time at CERN, especially at the times when the project's difficulty seemed to be overwhelming. Furthermore, I would like to acknowledge the technical assistance and advice given by several IT-CDA and RCS-SIS webmasters at CERN, especially Harris Tzovanakis and Salome Rohr. My last and greatest thanks go to Professor Kasper Knop Rasmussen for offering to supervise my BSc thesis, helping me obtain my bachelor's degree. Their assistance was critical to the success of this project.

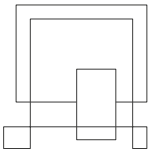
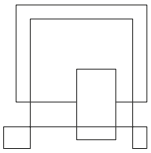
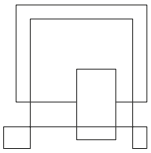
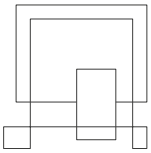


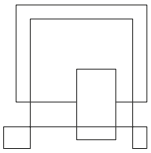
Table of content





List of figures and tables

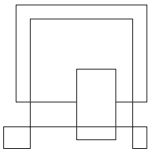




Abstract

You will be guided through the structure of the team, the project requirements, and how these were fulfilled through careful analysis. Furthermore, it will get into technical details how it was designed and implemented as well as tested during the process.

This report will also provide a summary of the project's results, conclusions, and future plans.



1 Introduction

CERN's Academic Training Lecture series include recordings and/or downloadable files of material allowing CERN users or interested parties to follow up these lectures through an online platform. The lectures are open to all members of CERN personnel (in particular staff members and fellows, associates, students, users, project associates and apprentices) free of charge. The complete catalogue of Academic Training lectures is **archived since 1968**, however, not all of them have videos.

This project is about **making the lecture views and searches more attractive**.

In this project, the *IT developer* (Franciska Török) and *RSC maintainer* (Harris Tzovanakis and his team) relied heavily on the *Project Initiator* (Maria Dimou), who collected the main requirements from the members of the *ATC*. During the process, several ATC meetings were held with the stakeholders (ATC) [1] and separate meetings between IT and RSC for the development of the website.

CDS [2] and Indico [3] are used to power the Academic Training website, which targets all CERN personnel, whose names are below referred to as *CERN Users*.

On the following rich picture, the solid arrows show direct impact while the dashed lines show dependency.

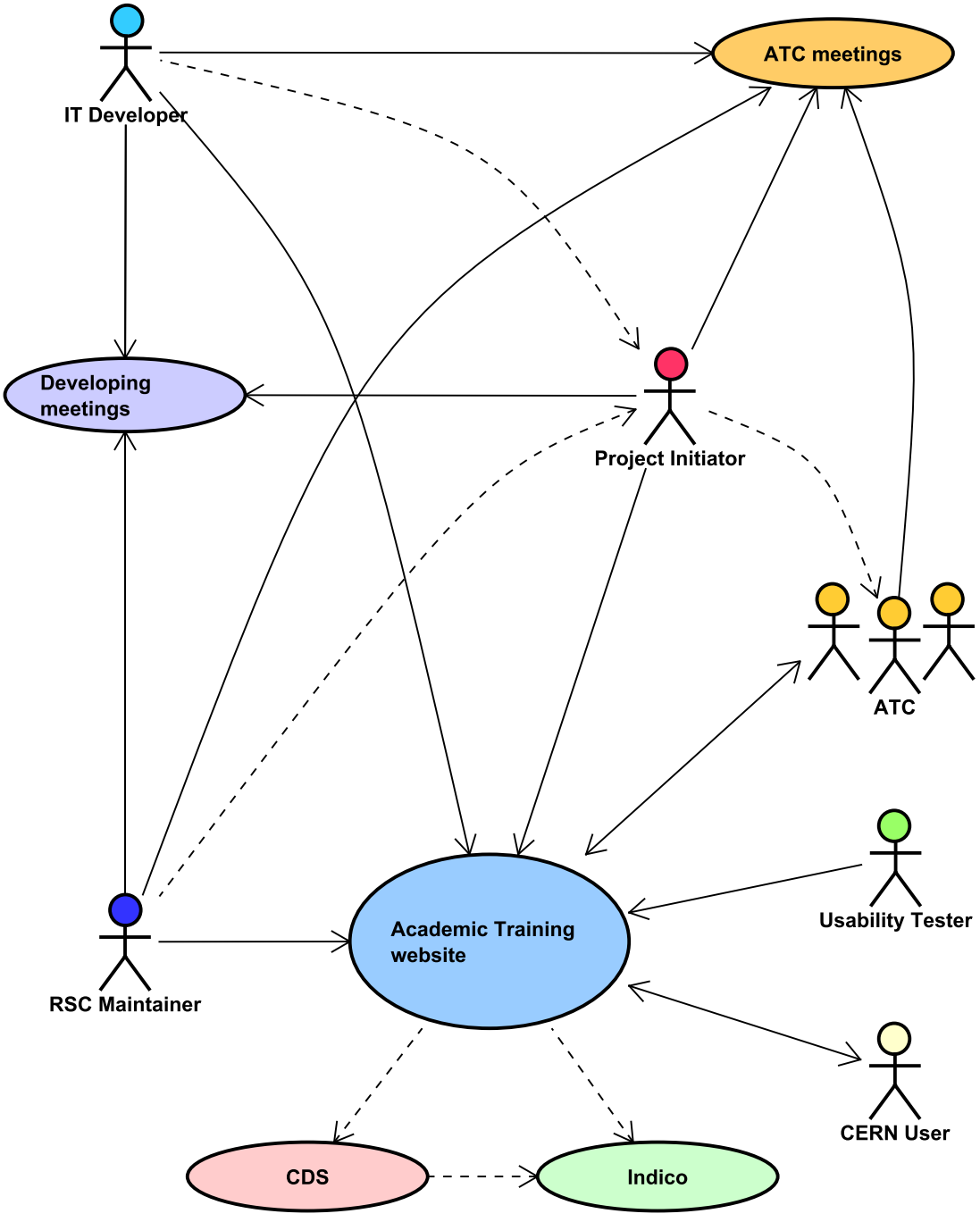
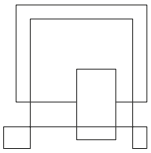


Figure 1: Rich Picture – The overall structure of the project



2 Analysis

Several requirements have been outlined by the ATC members for this project, which will be discussed in more detail below.

2.1 Actor descriptions

Different types of users will have access to different features and functionality as described in the actor descriptions.

User – User is an actor who uses the platform and performs several actions like watching the video lectures, searching for lectures, accessing lecture files, having the ability to contact the ATC, etc. The user does not have to be registered or logged in in order to perform these actions.

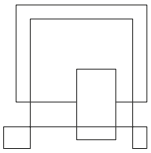
Administrator – The Administrator is the maintainer of the platform who makes sure that the site works as expected and takes responsibility for any changes that arise, such as the need to harvest new lectures or to change members of the ATC.

2.2 Requirements

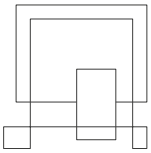
The following requirements are based on the problem defined in the Project Description [4], mandatory elements from CERN's design guidelines [5] and general regulations for the CERN Logo usage [6].

2.2.1 Functional Requirements

1. As a user, I want to see the lectures displayed by *thumbnail picture*, *speaker* and *date*, so I can watch the lecture that I want.
2. As a user, I want a search bar, so I can search lectures by *title, speaker, date, abstract, sponsor* and *keywords*.



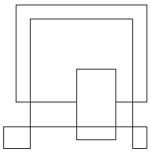
3. As a user, I want to see the search bar permanently, so I can search for a lecture whenever I want.
4. As a user, I want the lectures displayed chronologically on the *Main* page and on the *Search* result page, so that I can see the most recent first.
5. As a user, I want a dropdown menu of *Relevance* that sorts the search results by score (*Most relevant*), by date (*Newest first* and *Oldest first*), so I can sort in the order I want to see them.
6. As a user, I want to see the number of results after each search, so I know how many lectures were found for my search term.
7. As a user, I want to see a message displayed when no results are found after a search, so that I know, I must change the search term, and look for something else.
8. As a user, I want a *Contact* button that navigates to the email address of the stakeholders (ATC members), so I can contact them, give feedback or send a suggestion for future lecture.
9. As a user, I want an *Events* button, so that I can access and see the Full programme of the Academic Training Lectures (Indico [3]).
10. As a user, I want a button that navigates to an *About* page, so I can see who the members of the ATC are.
11. As a user, I want the *title, speaker, date, event details, abstract* (description of the lecture), *duration* and *sponsor* displayed for each lecture, so I can see the details of it.
12. As a user, I want a permanent header on every page top to find the *About, Contact,* and *Events* links, so I do not have to search or scroll for it for too long.
13. As a user, I want a *Help* tooltip next to the search bar, so I know how to use the search engine the most efficient way as possible.
14. As a user, I want to be able to force exact search matches, so I can find the precise lecture that I want.
15. As a user, I want the software to be able to display also *two-channel lectures*, so I can see one for the recording of the speaker and one for the slides used to illustrate the lecture.



16. As a user, I want a *list of files* displayed on the page of a lecture, if video is not present, so I can access and view non-video lectures as well.
17. As an administrator, I want to be able to harvest new lectures from the Academic Training Lecture series, so I can keep the software up to date.
18. As an administrator, I want to be warned if the harvesting fails, so I can avoid risks of the site being out of date.
19. As an administrator, I want to be able to include additional details from the event of the lectures (Indico), if it is not present already in the CDS, so I do not miss details that are needed by the stakeholders.
20. As an administrator, I want to be able to update the ATC members, so I can display an up-to-date information about the members and the Committee mission.

2.2.2 Non-Functional Requirements

1. The software must be publicly accessible on the internet.
2. There must be a logo specific to THIS software that gives the brand of the ATC.
3. The Academic Training lectures must be accessible separately from the collection on CDS.
4. The software must display mixed results. While the majority of the lectures are recorded and stored as videos, the old records are archived in PDFs, so they must also be managed.
5. The lecture harvesting should be automatic, performant (compare video loading speed on the site and in CDS; ensure no overhead appears due to the site views/scripts), and persistent to eventual CDS changes.
6. The software must use cold colors (ex. blue, black, white), as this is the general color scheme of CERN [7].
7. The site views, links and navigation must be user-friendly.
8. CERN is bilingual (French and/or English). This software must display views in English, because all the lectures are in English.
9. The software must belong to the domain name *.cern.ch*.



10. The software must be designed responsive, so that it can be viewed on smaller screens too, like on tablet and/or mobile devices. Since it is not possible to test it on every single device in the world, it is important to have responsive layout on some of the most popular devices (like iPhone 8, Samsung Galaxy S8, Google Pixel and Nexus 7).
11. The software must be usable from multiple browsers: Google Chrome, Firefox, Samsung Internet, Safari, Microsoft Edge, and Opera.
12. In order to ensure future maintainability, the software must make use of technologies in which CERN and the community have expertise.
13. The CERN toolbar must be displayed at the top of the software in the standard way, including a link to the homepage of CERN [8], to Sign In and to the CERN Directory.
14. The CERN logo [6] should always be used in the footer and be an active link to the homepage of CERN.
15. The CERN logo should *never* be used at the top left corner or in the header.
16. The CERN logo should *not* be animated or have states or scripted interactions.
17. The CERN logo should be *no* less than 60px wide, and its correct proportions must be respected.
18. A descriptive documentation site must be created and written for future maintenance.

Note: Regulations for optional CERN elements are not stated here.

2.3 Use Case Diagram

The following use case diagram represents use cases based on the requirements above.

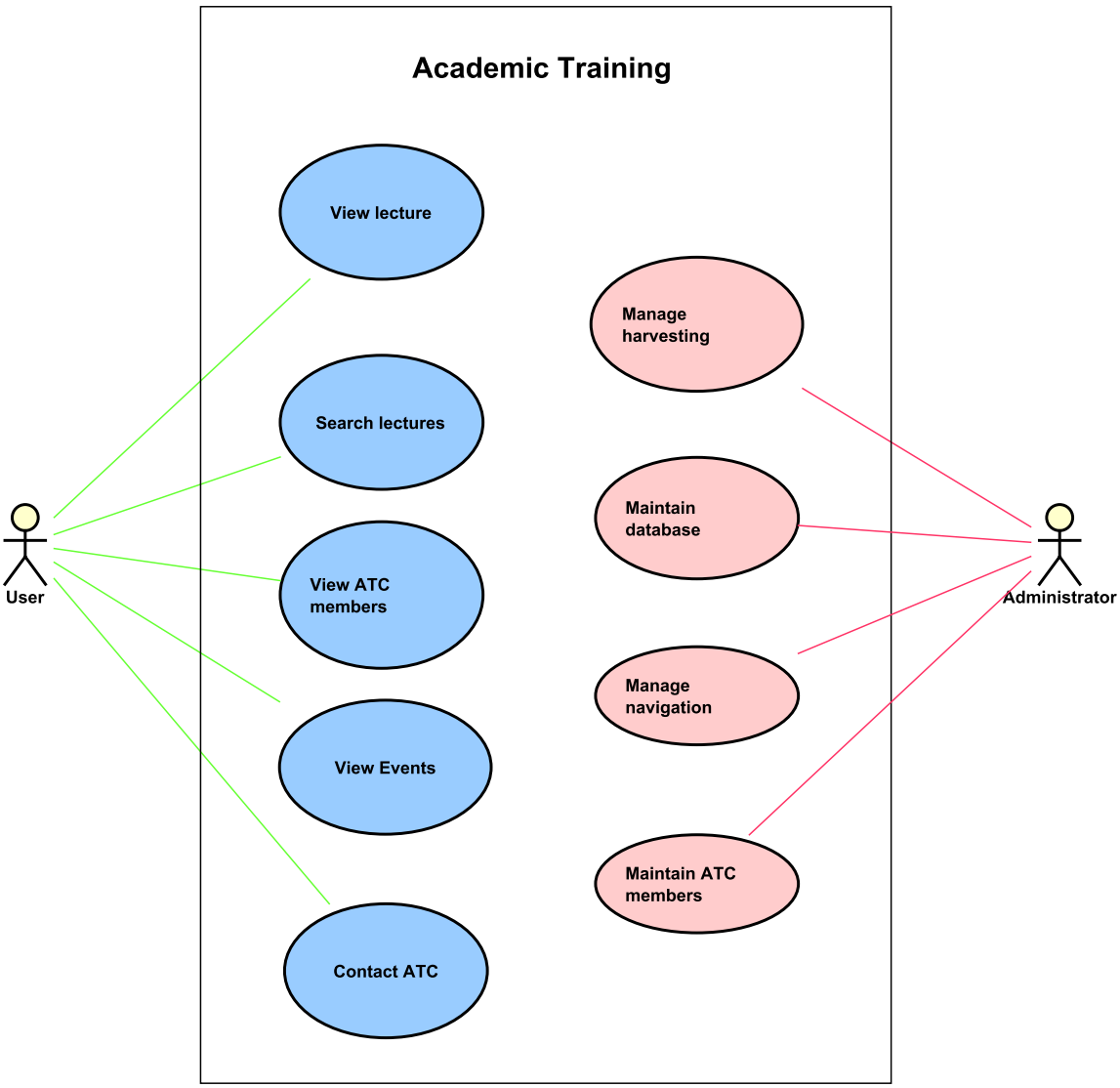
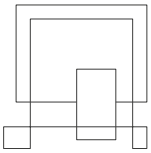


Figure 2: Use Case Diagram

Note: For more visible Use Case diagram see Appendix D.

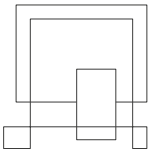


2.4 Use Case Descriptions

Use case	View lecture
Summary	View lecture and its details (title, speaker, date, abstract, event details, sponsor, keywords)
Actor	User
Precondition	• The lecture event exists in Indico • The lecture has taken place • The recording has been uploaded to CDS • The harvesting from CDS was successful • The connection to Indico API was successful
Postcondition	Harvested from CDS
Base sequence	1. Click on a lecture to watch 2. System talks through HTTP request to the REST API to ask for the called lecture 3. System checks if the lecture exists in the database 4. System sends back an HTTP response 5. System checks if the lecture is a single video, a two-channel video, has files or none. 6. System displays the lecture with its details accordingly
Note	If CDS is down, the videos and files cannot be reached (!) – an error message will be displayed that will originate from CDS. Everything else has a copy in the system's separate database (title, speaker, date, abstract, event details, sponsor etc.) that are not dependent on CDS and can be displayed regardless.

Figure 3: Use Case Scenario 1 – View lecture

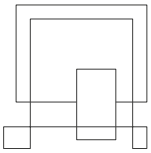
Use case	Search lectures
Summary	Search lectures and/or browse from search results
Actor	User



Precondition	Lectures exist in CDS
Postcondition	Harvested from CDS
Base sequence	1. Click on the search bar to search 2. Fill the search bar with content 3. Hit Enter OR click on the Search Icon 4. System checks if the search value in the Search bar has content a) If not, the system generates an empty search and displays all the lectures available b) If yes, go to step 5 5. System sets the search term with the searched value 6. System navigates to the search route after the query and displays the first page of the results
Note	- Empty search lists all the lectures available. - A guide (?) to help how to search is available next to the search bar: *You can search by title, speaker, date, abstract, sponsor and keywords.* *The default search operator is AND.* *For an OR search, use | between words:* *ex. particle | physics* *For exact match, use double quotes, like this:* *"dark matter"*

Figure 4: Use Case Scenario 2 – Search lectures

Use case	View ATC members
Summary	View the members of the Academic Training Committee (ATC) and their mission
Actor	User
Precondition	Maintainer has the latest ATC members deployed
Postcondition	-
Base sequence	1. Click on the 'About' button to view the members of the ATC



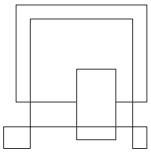
	a) In the header b) In the footer 2. System navigates to the site of the 'About Us' that displays the description of the ATC and the current members of it
Note	-

Figure 5: Use Case Scenario 3 – View ATC members

Use case	View events
Summary	Navigate to and view the full programme of the CERN Academic Training Lecture series
Actor	User
Precondition	Indico category 72 is accessible (Academic Training Lecture Regular Programme)
Postcondition	-
Base sequence	1. Click on the 'Events' button to view the full programme of the CERN Academic Training Lecture series a) In the header b) In the footer 2. System navigates to Indico
Note	If Indico is down, the full programme cannot be reached. However, the rest of the system still can be used.

Figure 6: Use Case Scenario 4 – View events

Use case	Contact ATC
Summary	Contact the ATC through a dedicated email address
Actor	User
Precondition	The *atc-contact@cern.ch* must be configured in the CERN e-groups
Postcondition	-
Base sequence	1. Contact the ATC members either:

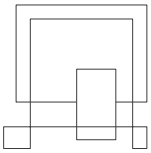


	a) Through the 'Contact' button on the header/footer OR b) 'Submit a suggestion for future topics' button 2. System opens the default email agent with the email address to the ATC
Note	Cancel can be done anytime.

Figure 7: Use Case Scenario 5 – Contact ATC

Use case	Manage harvesting
Summary	Harvest lectures from CDS
Actor	Administrator
Precondition	Lectures exist in CDS
Postcondition	OAI-PMH mechanism for MARCXML translation
Base sequence	1. Run the backend 2. Login in the database with admin credentials 3. Add lecture 4. Migrate lectures a) in case of harvest failure, send an email to the Academic Training site admins' e-group
Note	- The lecture's details are retrieved from a MARCXML file that needs conversion before harvesting the records from CDS to the database. This is necessary, because MARCXML has data fields with unique numbers that correspond to a tag, and subfields that correspond to a character. Since it is not easily readable, it must be translated. That is the OAI-PMH mechanism (translation from XML to JSON). - The information that does not exist on CDS is obtained from the lecture's Indico event (Indico API).

Figure 8: Use Case Scenario 6 – Manage harvesting

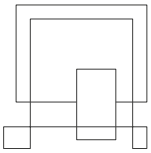


Use case	Maintain database
Summary	Manage database once change occurs or update is required
Actor	Administrator
Precondition	Lectures harvested
Postcondition	Admin logged in, git repository cloned
Base sequence	1. Run the backend 2. Login in the database with admin credentials 3. Add/modify lecture fields 4. Migrate lectures a) in case of harvest failure, send an email to the Academic Training site admins' e-group
Note	The abstract must be cleansed because it might have features that are not needed (like HTML tags or CSS attributes).

Figure 9: Use Case Scenario 7 – Maintain database

Use case	Manage navigation
Summary	Manage navigation of the project to render the app at different URLs
Actor	Administrator
Precondition	-
Postcondition	Admin logged in, git repository cloned
Base sequence	1. Add path(s) / URLs 2. Define route(s) of the path(s) 3. Add outlet to path
Note	A default blank page is added in case of a route that has no endpoint (it leads nowhere).

Figure 10: Use Case Scenario 8 – Manage navigation



Use case	Maintain ATC members
Summary	Maintain the ATC members on the About Us page, update members in case of change
Actor	Administrator
Precondition	The ATC chairperson informs the Administrator
Postcondition	About Us page exists
Base sequence	1. Take the list of ATC members 2. Update or add a new member by the following: a) key b) name c) profile d) position e) department 3. Update About Us page 4. Save changes
Note	The profile pictures of the members must be downloaded and added to the project repository.

Figure 11: Use Case Scenario 9 – Maintain ATC members

2.5 Domain Model

The Domain model presented below describes the Domain concepts and their relations. Users can view the ATC members, view lectures and/or search for these lectures, however, the Administrator can edit the ATC members, harvest these lectures and/or modify their content. The lectures are being harvested from CDS and some additional data is extracted from the lecture's Indico event (such as sponsor and keywords).

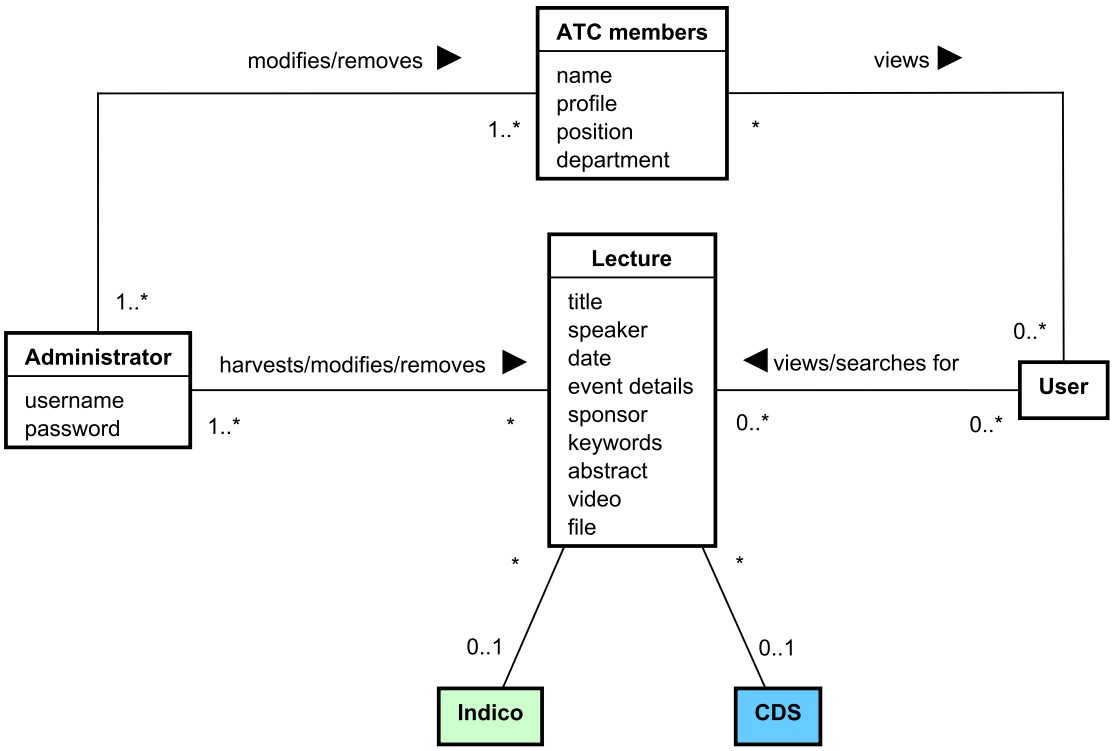


Figure 12: Domain model

Note: For more visible Domain Model see Appendix D.

3 Design

The system's main goal is to serve the users with Academic Training lectures, that are easily accessible and pleasant to watch from a user-friendly site.

3.1 Architecture

This chapter will navigate through the architecture of the project.

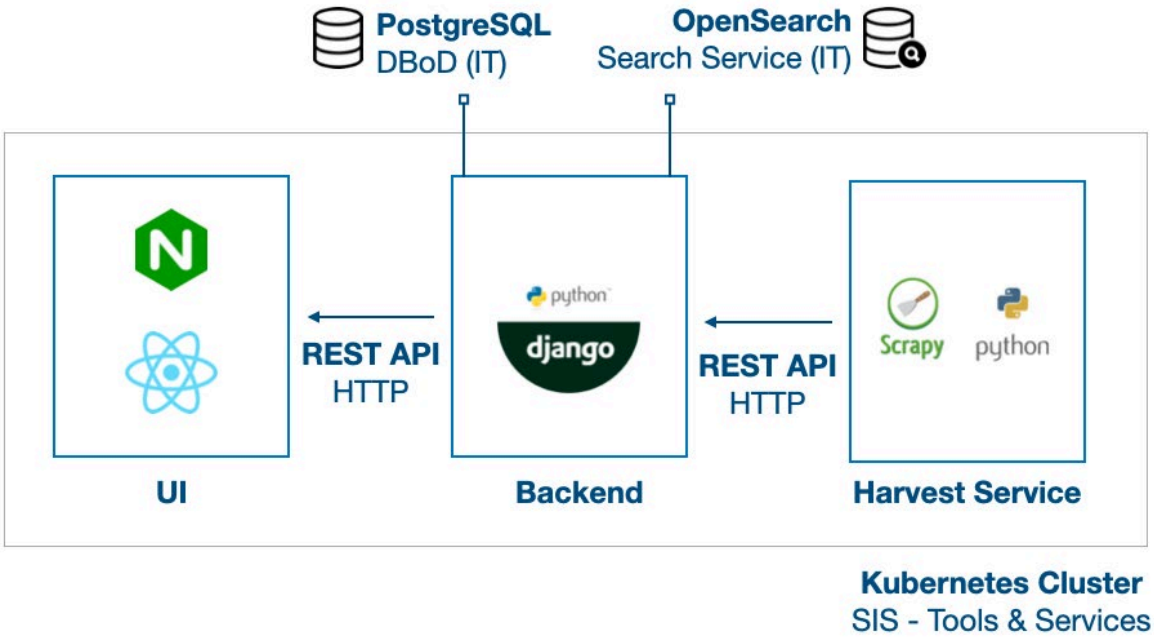
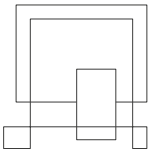


Figure 13: Architecture

CERN SIS - Tools & Services uses its general **Kubernetes** [9] **Cluster** for the Architecture.
As can be seen above, the whole architecture consists of **three main services**.
These three main services communicate through *HTTP requests* maintained by **REST API** that sets in the Backend.



3.1.1 Harvest

Initially, the CERN Academic Training lectures are collected from CDS. In order to harvest records from CDS, OAI-PMH mechanism has been used.
An already existing harvesting method was reused from **INSPIRE** [10] which contains already the data transformation from XML to JSON format.
When the harvest is successful, the Backend's REST API is used to create new entries.
Then harvesting procedure is running weekly to get the latest records from CDS.

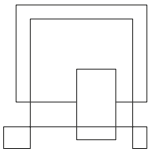
3.1.2 Backend

The lectures that are being harvested from CDS are stored in a **PostgreSQL** [11] database (**DBoD** [12]). Another service provides the **Search** of the website (**OpenSearch**) [13].

Note:
Some important data are not present in CDS. They are retrieved from the **Indico API**. We will get back to the reasons and details later.

3.1.3 UI

React manages the frontend part of the site and its aesthetic design as the main provider of the **UX**.
To facilitate long-term maintenance, the **Ant Design** [14] components were chosen for the UI framework, for which expertise at CERN already exists.



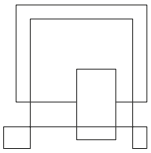
3.2 Technologies

- **Harvest:** Built by **Scrapy** [15] and **Python** [16].
- **Backend:** Built by **Django REST framework** [17] with **Python**.
- **UI:** Built by **React.js** SPA [18] and **Node.js** [19].

3.2.1 Technology alternatives

At the very beginning of the project, we have evaluated the technologies used at CERN in order to defend the solution we would eventually conclude on.

Frontend	Backend	Advantages	Disadvantages
Drupal [20]		Easier maintenance, deployment, authentication, data storage provided by CERN	Only a few lines of code would have been written which would have provided minimal educational value, since most things are already implemented + not much to write in the thesis + detrimental for long-term personal career
React	Django	Provides preferable educational value, the thesis can be written properly + technologies are advantageous for long-term personal career on the developer's side	After finishing the Technical Studentship contract at CERN, the project must be maintained by someone else + it is custom (must be built from scratch) and it is more complicated



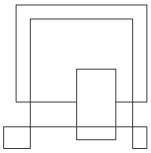
Wordpress [21]		It is very easy for non-experienced developers to create sites	It is not supported at CERN; it is not sophisticated enough for the harvesting
Invenio [22]		It is a CERN hosted application, there are experts at CERN	It is in early development and has the same situation as Drupal

Figure 14: Technology alternatives table

For technology alternatives, **Drupal** was one of the main suggestions for creating this platform, due to its popular use across CERN websites. Drupal is a content-management framework that offers customisation and flexibility. It helps to modify websites and content. The home website of CERN is also made by Drupal. CERN has a themed template which can be customized with the developer's choice.

While Drupal would have been easier to maintain, deploy, authenticate, and store data, the development interest would be minimal. Only a few lines PHP code would have been needed, leading only to a lesser technical experience gaining during the project. Since most technical aspects are already implemented, there would have been not much to do, and it would have had minimal educational value. Moreover, the maintainers' team has experience with the technology eventually used, so they adopted the chosen solution, which was **React-Django**.

The other options of the table were less considered, and the reasons are on the table itself. The reason why Drupal was extensively evaluated is that there are hundreds of CERN websites built with Drupal.



3.3 ER Diagram

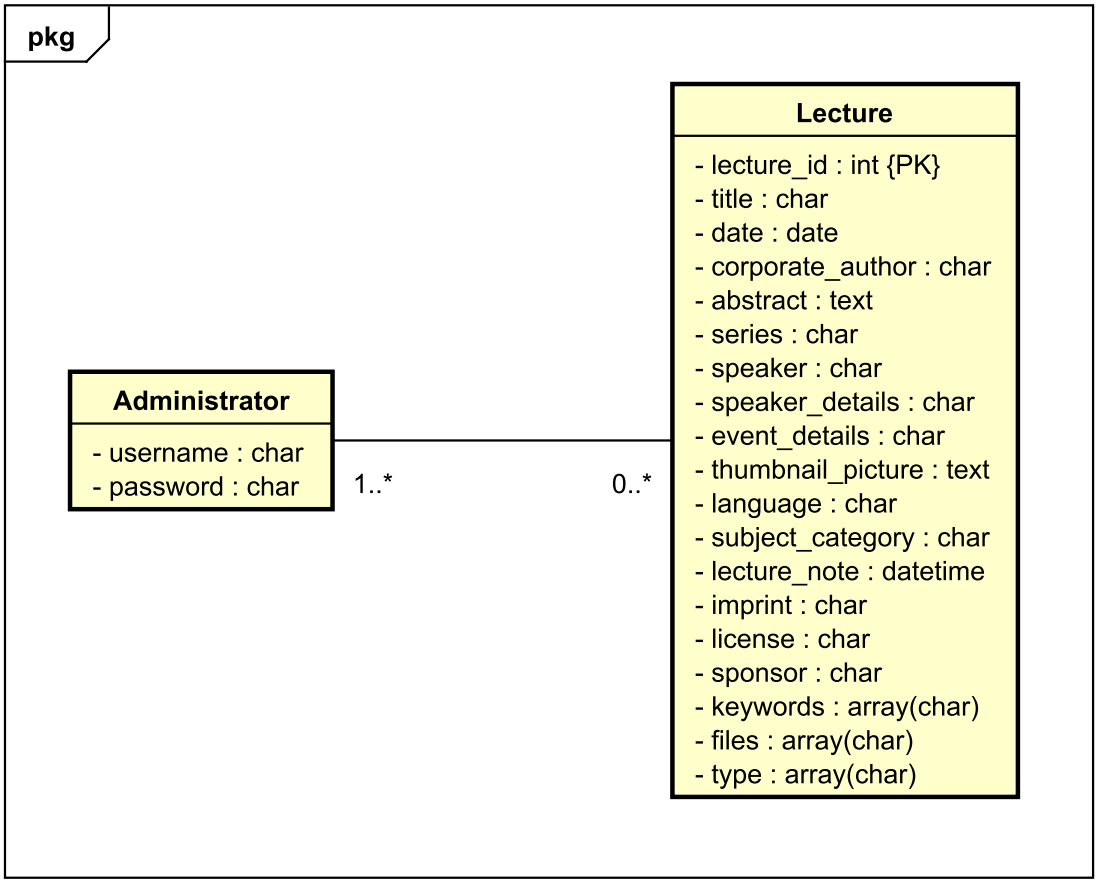
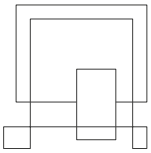


Figure 15: ER diagram



3.4 Tools

Next to the tools that have been used will indicate their purpose in this project:

- **Adobe XD** ® – UI/UX design
- **GitHub** – managing git repositories, team collaboration and code development
- **GitLab** – site documentation
- **Node.js**
- **Visual Studio Code** – writing code
- **JIRA** – project management, backlog, and workflow tracking
- **Docker** – packaging the application and dependencies into a container
- **Stack Overflow** – debugging and looking up solutions for bugs

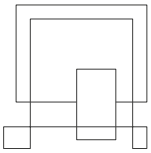
3.5 Package Manager

For this project the used package manager has fallen to ***yarn*** [23], which is a software packaging system developed for Node.js JavaScript runtime environment that provides speed, consistency, stability, and security as an alternative to ***npm*** [24]. There was no specific reason to pick one over the other.

3.6 Containerization

This software can be delivered more quickly using **Docker** [25] as it separates applications from their infrastructure. This platform provides the management of the **container lifecycles**, which are the following:

- Developing the application and its supporting components using containers.
- Transforming the container to the unit that distributes and tests the application.
- Deployment of the application into the production environment, as a container.



This application requires some services to run, such as the database and Elasticsearch. The following files are configured in **Docker** and **Docker-Compose** to run these services cross-platform and conveniently:

Dockerfile

This Dockerfile builds a fully functional image of your application with all of the static assets it requires.

docker-compose.yml

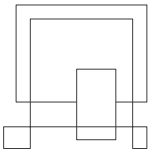
This file contains and exposes locally the minimal set of service containers needed for developing the instance locally:

`db`: The database, PostgreSQL, exposing the 5432 port
`es`: Elasticsearch version 7, exposing the 9200 and 9300 ports

When developing and running your instance locally these services can be accessed by the application.

3.7 UI design choices

As a direct relation to the UI, only User use cases will be presented. Please find the UI design choices in **Appendix I**.



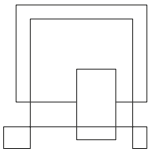
3.8 Navigation

The navigation of the project has been accomplished with **React Router** [26] which is a client and server-side routing library for React, a JS library for building user interfaces.

The used routes have a purpose to lead us to separate URLs of our website, like the main page, the detail page of a lecture, the search results and the page that displays the members of the ATC.

The routes that are needed for the site are defined with their paths the following way:

- **Main page** – "/"
- **Search results page** – "/search"
- **About Us page** – "/about-us"
- **Collection of lectures** – "/lectures"
- **Lecture page** – "/lectures/:lectureId"
- **Blank page** (in case of a route that has no endpoint & leads nowhere) – "/*"



4 Implementation

4.1 View lecture functionality

First of all, we create a connection with the API through **Axios** [27].

/ui/src/api/api_root.tsx

```
import axios from "axios";

const api = axios.create({
  baseURL: `/api/v1/`,
});

export function getApiRoot() {
  return api;
}
```

The code below fetches the lectures from the API root.

/ui/src/routes/lecture.tsx

```
const fetchLecture = async () => {
  try {
    ...
    const results = await getApiRoot().get(`/lectures/${lectureId}/`);
    setLecture(results.data);
    ...
  } catch (error) {
    setLecture({});
  }
};

useEffect(() => {
  fetchLecture();
}, [lectureId]);
```

4.1.1 View video lecture

It takes the type that is a `video`.

/ui/src/routes/lecture.tsx

```
const isVideo = lecture.type && lecture.type.includes("video");
```

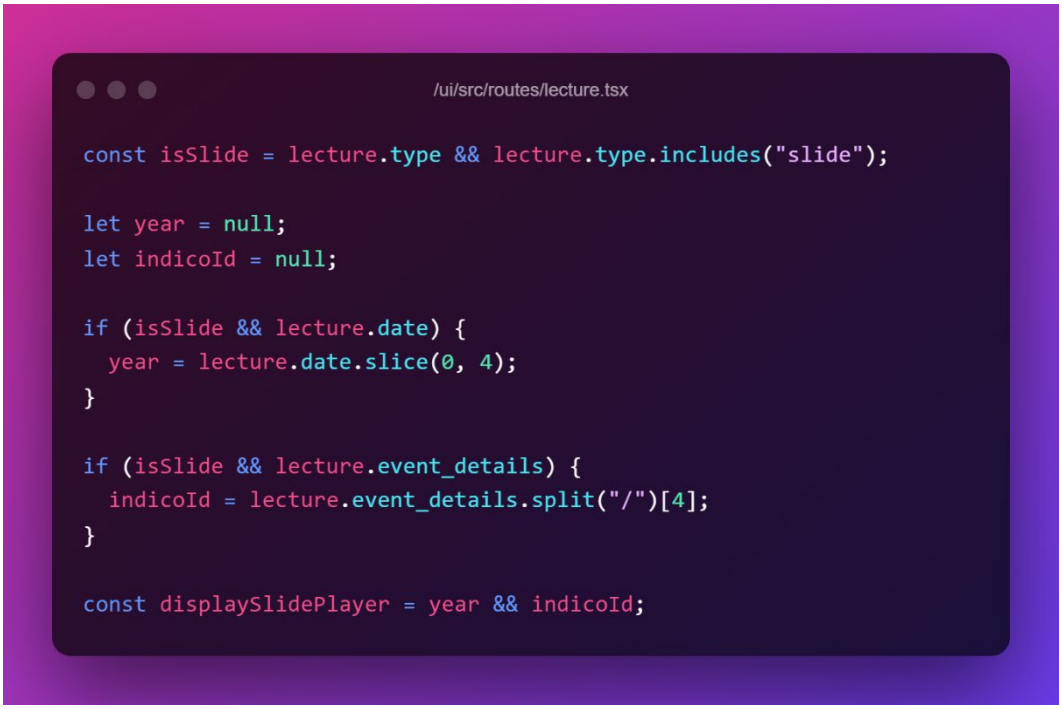
Then if it is a single video, it displays the following way:

/ui/src/routes/lecture.tsx

```
{
isVideo && (
    <div className="video-window">
      <iframe
        title={lecture.title}
        src={`https://cds.cern.ch/video/${lectureId}?showTitle=true`}

        allowFullScreen
      />
    </div>
  );
}
```

4.1.1.1 View two-channel video

It takes the type that is a `slide`. This one needs the `year` and the `indicoId` in order to work. The `year` is taken from the `date` while the `indicoId` is taken from the Indico event of the lecture (`event_details`).



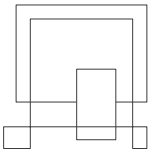
/ui/src/routes/lecture.tsx

```
const isSlide = lecture.type && lecture.type.includes("slide");

let year = null;
let indicoId = null;

if (isSlide && lecture.date) {
  year = lecture.date.slice(0, 4);
}

if (isSlide && lecture.event_details) {
  indicoId = lecture.event_details.split("/")[4];
}

const displaySlidePlayer = year && indicoId;
```

Then if it is a two-channel video, it displays the following way:

/ui/src/routes/lecture.tsx

```
{
  displaySlidePlayer && (
    <div className="video-window">
      <iframe
        title={lecture.title}
        src=
{`https://mediastream.cern.ch/MediaArchive/Video/Public2/weblecture-
player/index.html?year=${year}&lecture=${indicoId}`}
        allowFullScreen
        scrolling="no"
        frameBorder="0"
      />
    </div>
  );
}
```



4.1.2 View non-video lecture

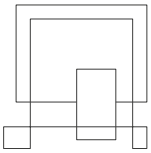
Some lectures have neither videos nor files. Those only display the leftover metadata like date, title, speaker, description, if any.

4.1.2.1 View list of files

This code checks if there are any files, then it displays the title to download files as well as the list of files.

/ui/src/routes/lecture.tsx

```
{
  lecture.files && lecture.files.length > 0 && (
    <div className="files">
      <div className="download-title">
        <DownloadOutlined
          style={{
            color: "#fff",
            fontSize: "250%",
          }}
        />
        <Title level={2}>Download files:</Title>
      </div>

      <List
        itemLayout="horizontal"
        dataSource={lecture.files || []}
        split={false}
        renderItem={(item: string, index) => {
          return (
            <List.Item key={index}>
              {index + 1}.{" "}
              <a title={item} rel="noreferrer" target="_blank" href={item}>
                {filenameFromUrl(item)}
              </a>
            </List.Item>
          );
        }}
      />
    </div>
  );
}
```



4.2 Search lectures functionality

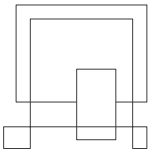
Let's begin from the backend. The backend has one of the most important roles of the search functionality.

/backend/cds/views.py

```python
class LectureViewSet(viewsets.ModelViewSet):
    # Calls all the lectures
    queryset = Lecture.objects.all()
    ...


class LectureDocumentView(DocumentViewSet):
    document = LectureDocument
    ...

    filter_backends = [
        FilteringFilterBackend,
        OrderingFilterBackend,
        DefaultOrderingFilterBackend,
        SimpleQueryStringSearchFilterBackend,
    ]
```

/backend/cds/views.py

```python
    # The default operator for search is AND
    simple_query_string_options = {
        "default_operator": "and",
    }

    # Fields for the search are defined here
    filter_fields = {
        "lecture_id": None,
        "type": None,
        "keywords": None,
        "series": None,
        "sponsor": None,
        "speaker": None,
        "subject_category": None,
    }
```



/backend/cds/views.py

```python
# Fields for the ordering are defined here
ordering_fields = {
    "lecture_id": None,
    "title": None,
    "date": None,
    "score": None,
    "series": None,
    "subject_category": None
}

# `boost` gives score to the variables,
# meaning that the most important ones are going to be displayed first
simple_query_string_search_fields = {
    "title": {"boost": 5},
    "abstract": {"boost": 1},
    "type": {"boost": 5},
    "keywords": {"boost": 6},
    "sponsor": {"boost": 10},
    "speaker": {"boost": 10},
    "series": {"boost": 10},
    "subject_category": {"boost": 10},
}

# Sorts by the score of the lectures and the date
ordering = (
    "_score",
    "-date",
)
```

4.2.1 Help for search

A tooltip has been added next to the search bar to help users how to search.

/ui/src/components/SEARCH_BAR.tsx

```
<Tooltip
  color="#0033a0"
  key="#0033a0"
  placement="bottomRight"
  title={
    <span style={{ whiteSpace: "pre-line" }}>
      <strong>How to search:</strong>
      {helpText}
    </span>
  }
  arrowPointAtCenter
>
  // The (?) icon
  <QuestionCircleOutlined />
</Tooltip>
```

4.2.2 View search results

First, the searched lectures are fetched from the API root.

/ui/src/routes/results.tsx

```
const searchLectures = useCallback(async () => {
    try {
      ...
      const response = await getApiRoot().get(`/search/lectures/`, {
        params: {
          ...
          search_simple_query_string: searchValue,
          ...
        },
      });
      ...
      setLectures(response.data.results);
      ...
    } catch (error) {
      setLectures([]);
    }
  }, [searchValue, ...]);
```

Thumbnails are filtered the following way:

/ui/src/routes/results.tsx

```
{
  // Lecture that has a thumbnail
  lecture.thumbnail_picture && (
    <div className="list-thumbnail">
      <img alt="thumbnail" src={lecture.thumbnail_picture} />
    </div>
  );
}
```

/ui/src/routes/results.tsx

```
{
  // Lecture that doesn't have a thumbnail
  !lecture.thumbnail_picture && (
    <div className="list-thumbnail">
      <div className="blank-thumbnail">
        <FileFilled
          style={{
            fontSize: "350%",
            opacity: "0.6",
          }}
        />
      </div>
    </div>
  );
}
```

4.2.3 View number of results

/ui/src/routes/results.tsx

```
...
const [total, setTotal] = useState<number>(0);
...
const searchLectures = useCallback(async () => {
    try {
      ...
      // This is where it counts the total amount of lectures (!)
      setTotal(response.data.count);
      ...
    }
  ...
});
...
<Content id="atc-content">
  ...
  <Title>
    {total} Search {pluralizeUnlessSingle('result', total)}:{" "}
    ...
  </Title>
  ...
</Content>
```

4.2.4 Sort results by relevance

/ui/src/routes/results.tsx

```
const [order, setOrder] = useState<SortOptions>(SortOptions.Default);

const searchLectures = useCallback(async () => {
    ...
      const response = await getApiRoot().get(`/search/lectures/`, {
        params: {
          ordering: order === SortOptions.Default ? undefined : order,
          ...
        },
      });
      ...
  }, [..., order]);
```

This component displays a dropdown menu for sorting.

/ui/src/routes/results.tsx

```
<SortMenu sortMethod={order} handleChange={setOrder} />
```

The options for the dropdown menu are included in the SortMenu component.

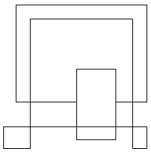
/ui/src/components/SORT_MENU.tsx

```tsx
export const SORT_MENU = ({
  handleChange,
  sortMethod,
}: {
  handleChange: (value: SortMethodType) => void;
  sortMethod: SortMethodType;
}) => {
  const { Option } = Select;

  return (
    <div className="sort-container">
      <Select value={sortMethod} onChange={handleChange}>
        <Option value="relevance" key="relevance">
          Most relevant
        </Option>
        <Option value="newest" key="newest">
          Newest first
        </Option>
        <Option value="oldest" key="oldest">
          Oldest first
        </Option>
      </Select>
    </div>
  );
};
```

Lecture interface not only exports the lectures, but the sort method type as well.

/ui/src/models/lectures.ts

```ts
export enum SortOptions {
  Default = "relevance",
  Oldest = "date",
  Newest = "-date",
}
```



4.2.5 No results

This component displays an empty container.

/ui/src/routes/results.tsx

```tsx
<Empty className="empty" description="No results found" />{" "}
```

4.3 View ATC members functionality

Card components have been used for displaying the ATC members.

/ui/src/routes/about_us.tsx

```tsx
const members = getMembers();
...
{
  members.slice(0, 1).map((member: any) => {
    return (
      <Space>
        <Col key={member.key} span={8}>
          <Card hoverable className="member-card">
            <Card.Grid className="grid-style">
              <div className="member-content">
                <Title level={3}>{member.department}</Title>
                <Avatar
                  size={120}
                  src={
                    require("../photos/members/profiles/Urs_Wiedemann.jpg")
                      .default
                  }
                />
                <Title level={2}>{member.name}</Title>
                <p>{member.position}</p>
              </div>
            </Card.Grid>
          </Card>
        </Col>
      </Space>
    );
  });
}
...
```

The members are defined in a separate JSON file as follows:

/ui/src/photos/members/members.ts

```
...
{
    key: 15,
    name: "Frank TECKER",
    profile: baseUrl + "Frank_Tecker.jpg",
    position: "",
    department: "BE",
}
...
```

4.4 Contact ATC functionality

All of them use a *mailto* identifier that automatically opens the default email agent of the user.

4.4.1 On the header

/ui/src/components/MENU.tsx

```
<Menu
  onClick={handleClick}
  mode="horizontal"
  className="menu"
  overflowedIndicator={false}
>
  ...
  <Menu.Item className="contact-us" key="contact-us">
    <Typography.Link href="mailto:atc-contact@cern.ch" target="_blank">
      <Title level={2} className="contact-us-link">
        Contact
      </Title>
    </Typography.Link>
  </Menu.Item>
</Menu>
```

Then it is built into the header as a part of the Menu.

/ui/src/components/AT_HEADER.tsx

```
{
// In case of a bigger screen
  (toggleMenu || screenWidth > 992) && <MENU />;
}
```

/ui/src/components/AT_HEADER.tsx

```
{
// In case of a smaller screen
  (toggleMenu || screenWidth <= 992) && (
    <Drawer
      placement="right"
      width="300px"
      className="drawer"
      onClose={toggleCollapsed}
      visible={!state.collapsed}
      destroyOnClose={true}
      closeIcon={<CloseOutlined style={{ color: "#fff" }} />}
    >
      <MENU />
    </Drawer>
  );
}
```

4.4.2 On the footer

/ui/src/components/CERN_FOOTER.tsx

```
<Menu className="menu nav">
  ...
  <Menu.Item>
    <Typography.Link href="mailto:atc-contact@cern.ch"
target="_blank">
      Contact
    </Typography.Link>
  </Menu.Item>
</Menu>
```

4.4.3 Suggestion link

/ui/src/components/CERN_FOOTER.tsx

```
<Typography.Link href="mailto:atc-contact@cern.ch"
target="_blank">
  <Title className="hover-underline-animation">
    Submit a suggestion for future topics
  </Title>
</Typography.Link>
```

4.5 Harvesting

The process of harvesting happens through a mechanism called **OAI-PMH** that in our project is used to translate XML files to JSON format. The fundamental data of an Academic Training lecture record is retrieved from CDS in MARCXML format.

4.5.1 OAI-PMH mechanism for MARCXML [28] translation

There are some tags with numbers in the XML that each of them corresponds to a specific variable, like the title, speaker, date, etc.
However, we cannot really see or understand them clearly from this metadata. Therefore, we shall take this metadata through a process that **translates our XML to a meaningful data** that is easily readable. In this case it is transformed to **JSON**. This process is the **OAI-PMH** mechanism that was mentioned earlier, and it is going to be described below. This method is reused from INSPIRE [10].

4.5.2 CDS Spider

The harvesting happens in a Scrapy file that collects data from CDS.
First, it builds the *CDS URL* with the query, size and field.

/harvest/harvest/spiders/cds_spider.py

```
def __build_cds_url(self, query, field=SEARCH_FIELD, size=SIZE):
        return CDS_URL.format(query=query, size=size, field=field)
```

The default URL with these fields included in the variable *CDS_URL*.

/harvest/harvest/spiders/cds_spider.py

```
CDS_URL = "https://cds.cern.ch/search?
ln=en&cc=Academic+Training+Lectures&action_search=Search&op1=a&m1=a&p1=&
f1=&c=Academic+Training+Lectures&c=&sf=&so=d&rm=&rg={size}&sc=0&of=xm&p=
{query}&f={field}"
```

It starts requests from CDS:

/harvest/harvest/spiders/cds_spider.py

```
def start_requests(self):
        if self.migrate_all:
            self.all_years_gen = self.__gen_all_years
            item = next(self.all_years_gen)
            url = self.__build_cds_url(item["year"], item["field"])

        else:
            query = "->".join(self.query)
            url = self.__build_cds_url(query)
        ...
```

This part selects records, then it tries to loop through them by getting the parsed records.

/harvest/harvest/spiders/cds_spider.py

```
def parse(self, response):
        sleep(3)

        response.selector.remove_namespaces()
        records = response.selector.xpath("//record")

        LOGGER.info("Harvested records", count=len(records))

        for record in records:
            try:
                sleep(3)
                yield self.parse_item(record, original=record.get())
            except Exception as err:
                LOGGER.error(err)
```

Here, to get all the records from CDS, it is done this way:

/harvest/harvest/spiders/cds_spider.py

```
try:
        if self.migrate_all and (item := next(self.all_years_gen)):
                LOGGER.debug("Harvesting next page", year=item["year"])
                url = self.__build_cds_url(item["year"], item["field"])
                LOGGER.debug("Harvesting url", url=url)
                yield Request(url, callback=self.parse)
        except StopIteration:
            LOGGER.debug("Harvesting all is finished.")
```

4.5.3 XML-JSON conversion

INSPIRE already has a method that transforms data from MARCXML to JSON and vice-versa (**INSPIRE-DoJSON**) [29]. This has been reused here.

It takes our XML from CDS and translates it to JSON with matching variables. For example, let's say we want to take the ID of the lecture which tag from the XML is 001. It will be assigned to a new record called *lecture_id* this way:

/harvest/harvest/spiders/cds_spider.py

```
def parse_item(self, selector, original=None):
        ...

        record["lecture_id"] = selector.xpath(".//controlfield[@tag=001]/text()").get()
```

Now here is where the data is passed to create new record. But first, note that these libraries must be imported:

/harvest/harvest/spiders/cds_spider.py

```
from dojson.contrib.marc21.utils import create_record
from inspire_dojson.cds import cds2hep_marc
```

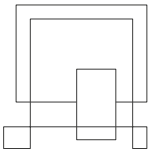
Last but not least, this part takes the original record and creates a new one by converting the XML record to JSON.

/harvest/harvest/spiders/cds_spider.py

```
data = cds2hep_marc.do(create_record(original))
...
return record
```

When the harvest shows a successful response, the Backend's REST API is used to create new entries.

Note: The harvesting procedure is running daily to get the latest records from CDS.



4.6 Maintain the database

The backend of the software is based on **Django REST Framework**.

After the installation of the Django REST Framework, the *'rest_framework'* is being added to INSTALLED_APPS.

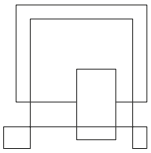
/backend/backend/settings/base.py

```
INSTALLED_APPS = [
    ...
    # external apps
    "rest_framework",
    "rest_framework.authtoken",
    ...
```

However, since our API is browsable, a file also configures the API's URL paths.

/backend/backend/urls.py

```
from accounts.views import UserViewSet
from cds.views import LectureDocumentView, LectureViewSet
from django.contrib import admin
from django.urls import include, path
from rest_framework import routers

# Default router
router = routers.DefaultRouter()

# Register users
router.register(r"users", UserViewSet)

# Register lectures
router.register(r"lectures", LectureViewSet)

# This one is for search
router.register(r"search/lectures", LectureDocumentView,
basename="lecturedocument")
```



The *urlpatterns* list routes URLs to views. The last one (*api-auth/*) is responsible for the REST framework's login and logout views.

/backend/backend/urls.py

```
urlpatterns = [
    path("admin/", admin.site.urls),
    path("api/v1/", include(router.urls)),
    path("api-auth/", include("rest_framework.urls",
namespace="rest_framework_auth")),
]
```

There is one configuration dictionary named REST_FRAMEWORK that contains all global settings for a REST framework API.

4.6.1 Fields in the database

The required fields are defined the following way:

/backend/cds/models.py

```
class Lecture(models.Model):
    lecture_id = models.IntegerField(unique=True, db_index=True)
    title = models.CharField(max_length=250)
    date = models.DateField(null=True, blank=True)
    ...
    etc.
```

4.6.2 Bleaching the abstract

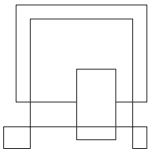
The abstract of the lecture **must be cleansed** before displaying on the UI, because CDS records might have some formatting styles (like HTML tags) that were initially meant for CDS, but they still exist in the metadata. Consequently, they disturb the current UI.

/backend/cds/models.py

```
def save(self, *args, **kwargs):
        try:
            self.abstract = clean(
                self.abstract,
                strip=True,
                tags=["p", "div", "strong", "span", "ul", "li"],
                attributes={"a": ["href"]},
                strip_comments=True,
            )
        except Exception:
            pass
        super().save(*args, **kwargs)
```

As can be seen above, the abstract is taken, strip will remove entirely invalid markups (strip_comments similarly with built-in comments), and then tags and attributes will only show elements that can be **allowed** within the abstract.

For this, a so-called **bleach** [26] Python package has been used.



5 Test

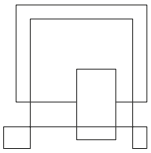
5.1 Test Specifications

5.1.1 Use case: View lecture functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Click on the lecture	Verify that the video is displayed, is playable and can be watched full screen. If it is a two-channel video, verify that it displays both the speaker and the slides of the lecture in two different windows within the video, is playable and can be watched full screen. If it is a non-video lecture that has files, verify that it displays the link to the files, they can be opened, and they are downloadable.	pass	5.1.1

5.1.2 Use case: Search lectures functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Click on 'Search a lecture…' search bar	If the search field is not modified - verify that the empty search retrieves all the existing lectures. If the search field is not null - verify that the searched term is not case-sensitive, and it displays lectures regardless.	pass	5.1.2
2	Click on "Search" button	If the title/speaker/abstract/sponsor/	pass	5.1.2



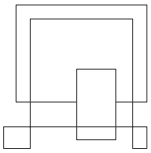
		keywords are already in database - verify that the searched term is displayed. If not all the values are typed - verify that the searched item can be still displayed.		

5.1.3 Use case: View ATC members functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Click on “About” button	Verify that the button navigates to the “About us” page, where it displays the members of the ATC. Verify that the displayed components (member cards) are consistent regardless of the choice of browser (Chrome, Firefox, Samsung Internet, Edge, Safari, Opera)	pass	5.1.3

5.1.4 Use case: View Events functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Click on “Events” button	Verify that the button navigates to the Full Programme of the Academic Training Lectures (Indico).	pass	5.1.4

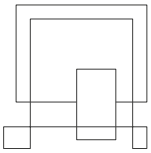


5.1.5 Use case: Contact ATC functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Click on "Contact" button OR "Submit a suggestion for future topics" button	Verify that it opens the user's default email agent with the ATC's email address.	pass	5.1.5

5.1.6 Use case: Manage harvesting functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Add/Update lectures	If the admin is logged in, add details of the lectures like title, speaker, abstract, sponsor and/or keywords, etc. Verify that lecture is added/updated.	pass	5.1.6
2	Migrate lectures	Collect lectures from CDS. Verify getting the harvested data. Migrate lectures from CDS to the website's API. Verify migration. If the data leaves behind missing fields (like sponsor and keywords), connect to Indico API, and verify that the additional data is retrieved. If harvesting fails, send email to the site admins.	pass	5.1.6

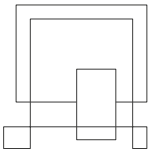


5.1.7 Use case: Manage database functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Modify lecture fields	If lectures are harvested from CDS and admin is logged in, modify lecture fields. Verify that lecture fields are modified.	pass	5.1.7
2	Migrate lectures	Verify the migration of lectures from CDS to the website's API. If the data leaves behind missing fields (like sponsor and keywords), connect to Indico API, and retrieve additional data. If harvesting fails, send email to the site admins.	pass	5.1.7

5.1.8 Use case: Manage navigation functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Define route(s)	Verify that the defined routes navigate to their exact destination (URL). If a route has no endpoint, verify that a default blank page is displayed.	pass	5.1.8



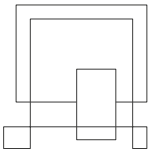
5.1.9 Use case: Manage ATC members functionality

Action no.	Action	Reaction	Pass/fail	VCRI
1	Add/Update ATC members	If the "About" page exists and the displayed data is outdated, add/update member by key, name, profile, position, department. Verify that the modification appears on the website.	pass	5.1.9

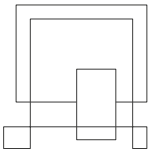
Possible test results:

Level	Description
1	Fatal
2	Critical
3	Major
4	Minor
5	Slight imperfections
6	Improvement

Test scenario	Action	Error ref	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
5.1.1	1							X
5.1.2	1							X
5.1.2	2						X	
5.1.3	1							X
5.1.3	1							X
5.1.4	1							X

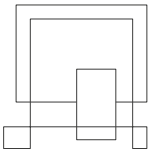


5.1.5	1							X
5.1.6	1							X
5.1.6	2							X
5.1.7	1							X
5.1.7	2							X
5.1.8	1							X
5.1.9	1							X



6 Results and Discussion

It was appreciated by the users, we got positive feedback, the project has perspectives for future expansions (see section 8 for details). There is of course a need to publicize, advertise the existence of the site, which only reached operational status in July 2022.



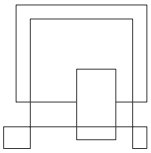
7 Conclusions

The technology evaluation and usage lead to a very good experience to the developer, they also integrated into an existing experience for the maintainer, and they produced a very attractive aesthetically and very functional technically website.

Future collaborations with the CDS team are discussed for other CERN lecture collections to be equally attractively promoted either via separate sites or from within the new CDS.

The photos that enhanced the site also gave the opportunity to understand better the life of the laboratory, the types of research, the experimental installations – also apply skills for photography.

There is a challenge now remaining to the ATC to evangelize the site to the user community.



8 Project future

The long-term infrastructure maintenance and evolution, the CDS harvesting supervision will be ensured by the team of Harris Tzovanakis, the CERN SIS - Tools & Services. ATC-related content updates will be done by Salome Rohr.

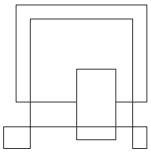
These have been agreed between Maria Dimou and the RCS directorate, project co-founder.

The website's views and especially searches depend on:

- the quality of the recordings,
- the creation of subtitles for the whole backlog of those lectures (after 1989), which contain video, and
- intelligent searches.

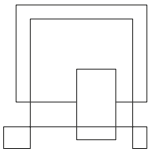
A new dedicated project is needed to harvest searchable strings from the subtitles and the 1968-1989 lectures, which do not contain video.

The CERN Login SSO will be part of the project in the early future, including the possibility to like lectures, add them to watched lists or favourites, and last but not least, a share option to popularize the Academic Training lecture series.

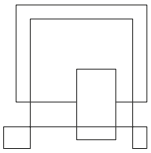


9 Sources of information

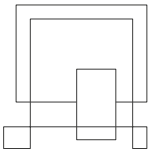
1. Academic Training Committee – About Us. [online] Available at: <https://academictraining.cern.ch/about-us/> [Accessed on 24 July 2022].
2. CERN Document Server (CDS). [online] Available at: <https://cds.cern.ch/> [Accessed on 22 July 2022].
3. Indico – Academic Training Lecture Regular Programme. [online] Available at: <https://indico.cern.ch/category/72/> [Accessed on 5 May 2022].
4. CERN Academic Training website – Project Description. [online] Available at: <https://it-student-projects.web.cern.ch/projects/cern-academic-training-web-site> [Accessed on 5 May 2022].
5. Guidelines for CERN websites. [online] Available at: <https://design-guidelines.web.cern.ch/guidelines/guidelines-cern-websites> [Accessed on 5 May 2022].
6. The CERN Logo. [online] Available at: <https://design-guidelines.web.cern.ch/guidelines/logo> [Accessed on 5 May 2022].
7. The CERN Colours. [online] Available at: <https://design-guidelines.web.cern.ch/guidelines/colours> [Accessed on 5 May 2022].
8. The Homepage of CERN. [online] Available at: <https://home.cern> [Accessed on 5 May 2022].
9. What is Kubernetes? [online] Available at: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/> [Accessed on 26 July 2022].
10. INSPIRE-DoJSON. [online] Available at: <https://github.com/inspirehep/inspire-dojson/> [Accessed on 26 July 2022].
11. What is PostgreSQL? [online] Available at: <https://www.postgresql.org/docs/current/intro-whatis.html/> [Accessed on 26 July 2022].



12. A new web application for the DBoD service. [online] Available at: <https://db-blog.web.cern.ch/blog/baptiste-legouix/2018-09-new-web-application-dbod-service/> [Accessed on 26 July 2022].
13. Introduction to OpenSearch. [online] Available at: <https://opensearch.org/docs/latest/opensearch/index/> [Accessed on 26 July 2022].
14. Ant Design of React. [online] Available at: <https://ant.design/docs/react/introduce> [Accessed on 26 July 2022].
15. Scrapy documentation. [online] Available at: <https://docs.scrapy.org/en/latest/> [Accessed on 26 July 2022].
16. Python. [online] Available at: <https://www.python.org/about/> [Accessed on 26 July 2022].
17. Django REST framework. [online] Available at: <https://www.django-rest-framework.org/> [Accessed on 26 July 2022].
18. React – Getting Started. [online] Available at: <https://reactjs.org/docs/getting-started.html/> [Accessed on 26 July 2022].
19. About Node.js. [online] Available at: <https://nodejs.org/en/about/> [Accessed on 26 July 2022].
20. What is Drupal? [online] Available at: <https://drupal-tools.web.cern.ch/docs/getting-started/getting-started> [Accessed on 26 July 2022].
21. Wordpress. [online] Available at: <https://wordpress.com/> [Accessed on 26 July 2022].
22. What is Invenio? [online] Available at: <https://inveniosoftware.org/about/> [Accessed on 26 July 2022].
23. Introduction – Yarn. [online] Available at: <https://yarnpkg.com/getting-started/> [Accessed on 29 September 2022].
24. What is npm? [online] Available at: <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/> [Accessed on 29 September 2022].
25. Docker overview. [online] Available at: <https://docs.docker.com/get-started/overview/> [Accessed on 26 July 2022].

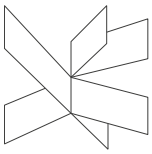


26. React Router. [online] Available at: <https://reactrouter.com/en/v6.3.0/getting-started/tutorial/> [Accessed on 29 September 2022].
27. Axios in React. [online] Available at: <https://www.geeksforgeeks.org/axios-in-react-a-guide-for-beginners/> [Accessed on 26 July 2022].
28. MARCXML. [online] Available at: <https://www.loc.gov/standards/marcxml/> [Accessed on 26 July 2022].
29. Bleach. [online] Available at: <https://pypi.org/project/bleach/> [Accessed on 26 July 2022].



10 Appendices

1. Appendix A – Project Description
2. Appendix C – Mockups
3. Appendix D – Diagrams
4. Appendix E – Source code
5. Appendix F – Source documentation
6. Appendix G – User Guide
7. Appendix I – UI Design Choices



BSc Thesis in Software Technology Engineering

Project Description

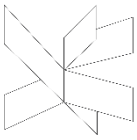
Building a website to promote the CERN Academic Training lectures

Franciska-Leonóra Török, 293171 IT

Supervisors:

Kasper Knop Rasmussen
VIA University College

Maria Dimou
CERN IT, Academic Training



VIA University College

10,368 characters

Software Technology Engineering

7th Semester

2023

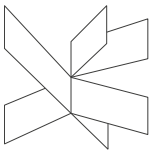
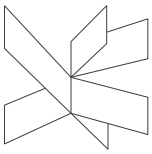
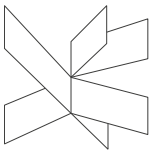


Table of content



List of figures and tables



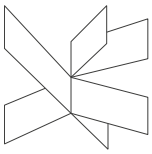
1 Background description

Lectures can be used by instructors to help students to easily acquire knowledge of terms, basic facts, and simple concepts. There is no difference in the effectiveness of lectures and other methods for conveying simple information [Bligh, 2000] [1].

CERN Academic Training lectures include both cutting-edge science and valuable historical information [2]. These lectures cover physics and technology research results, as well as news from other disciplines. Past lectures often present a great historical value. Lectures are open to all members of the CERN staff (including fellows, students, workers, users, and apprentices) free of charge. Some of these lectures are also available in a dedicated YouTube playlist [3]. All lectures are recorded and published on the Web, along with visual support materials. The complete catalogue of the Academic Training Programme lectures is archived since 1968, however, not all of them have videos. Similar lectures can also be found in other CERN programmes, such as Colloquia, Seminars, and various student programmes. Our project is only concerned with Academic Training lectures; however, it is a pioneering example for adoption by the others.

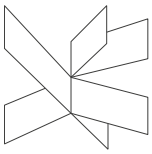
CERN Academic Training consists of a series of a few hourly lectures, all on consecutive days. Everyone at CERN receives an email inviting them to participate. It is generally decided by the lecturer and his/her sponsor what degree of complexity the lectures will have. On top of that, all lectures in any domain with their full content and their slides and video in the CERN Document Server (CDS) [4] are available for viewing from the lectures' description page [5]. Sponsors of these lectures are mostly members of the CERN Academic Training Committee (ATC).

While CDS ensures that these lectures are archived in a dedicated collection, there is no easy way to find lectures in that collection as they are simply indexed with the most recent first.



CDS has other purposes other than archiving Academic Training lectures, so their layout does not focus on aesthetics - it is only designed to store information.

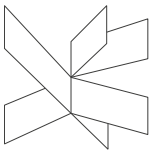
On the other hand, the reward for the prestigious speakers of Academic Training - who are willing to give their lectures almost free or with some symbolic financial contribution - is far from adequate. Lectures related to Academic Training are not widely advertised, and the speakers deserve more than just recording and storing their presentations.



2 Problem Statement

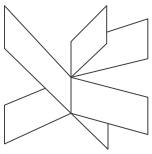
Even though CDS archives Academic Training lectures, there is no attractive presentation and advertising of the lectures to encourage the programme exploration by various target groups.

1. How to make the process of promoting lectures more attractive?
2. How to encourage the target audience to consume lectures more often?
3. How to help the target audience reach Academic Training lectures more easily?
4. How to gain proposals from target audience to get ideas for future lectures?
5. How to support speakers to promote their lectures?



3 Definition of purpose

In this project, the aim is to promote CERN's Academic Training lectures in a coherent way, and one that can be easily maintained in the future.



4 Delimitations

1. The product will prioritize videos over records with other filename extensions like PDFs from the lecture collection. Only videos will be displayed.
2. Due to the absence of subtitles/captions in CERN Academic Training lectures, searching between the lectures is only possible by using the archived metadata. If one could use the transcript of the whole lectures' content, the search results would be more sophisticated and more accurate. However, the transcription is not yet available, therefore, this is *not* going to be covered in this project.
3. Records have no category information encoded in their metadata. The categorization of lectures is not feasible due to the impossibility of providing a curator of the data that can serve for fetching relevant lectures. Amongst the over 900 records stored on CDS, no one would watch all the videos of the past to identify categories and subcategories such as Particle Physics, Technology, Computing, Science & Society. As a result of all these unavoidable reasons, the categories *cannot* be implemented in this project.
4. Allowing comments under lectures was a requirement from the project initiator up until the ATC members on the first call for feedback on the site decided *against* it due to the exposure to resentful and malicious messages. Such comments usually come from a skeptical point of view towards scientific evolution and technology. They occur even today under CERN social media accounts, like Facebook or Instagram.
5. Keywords is the best field we have for good quality search results. Unfortunately, hardly any lecture record contains keywords. This enhancement is time-consuming, can only be done by the specialists in the ATC and the development part of this project *cannot* help in any way.

5 Methodology

For this process will be used a quite unique methodology that incorporates the best of both Waterfall and Agile methods.

Agile-Waterfall hybrid method allows teams developing software to work within the Agile methodology, while the project manager sticks to the Waterfall approach.

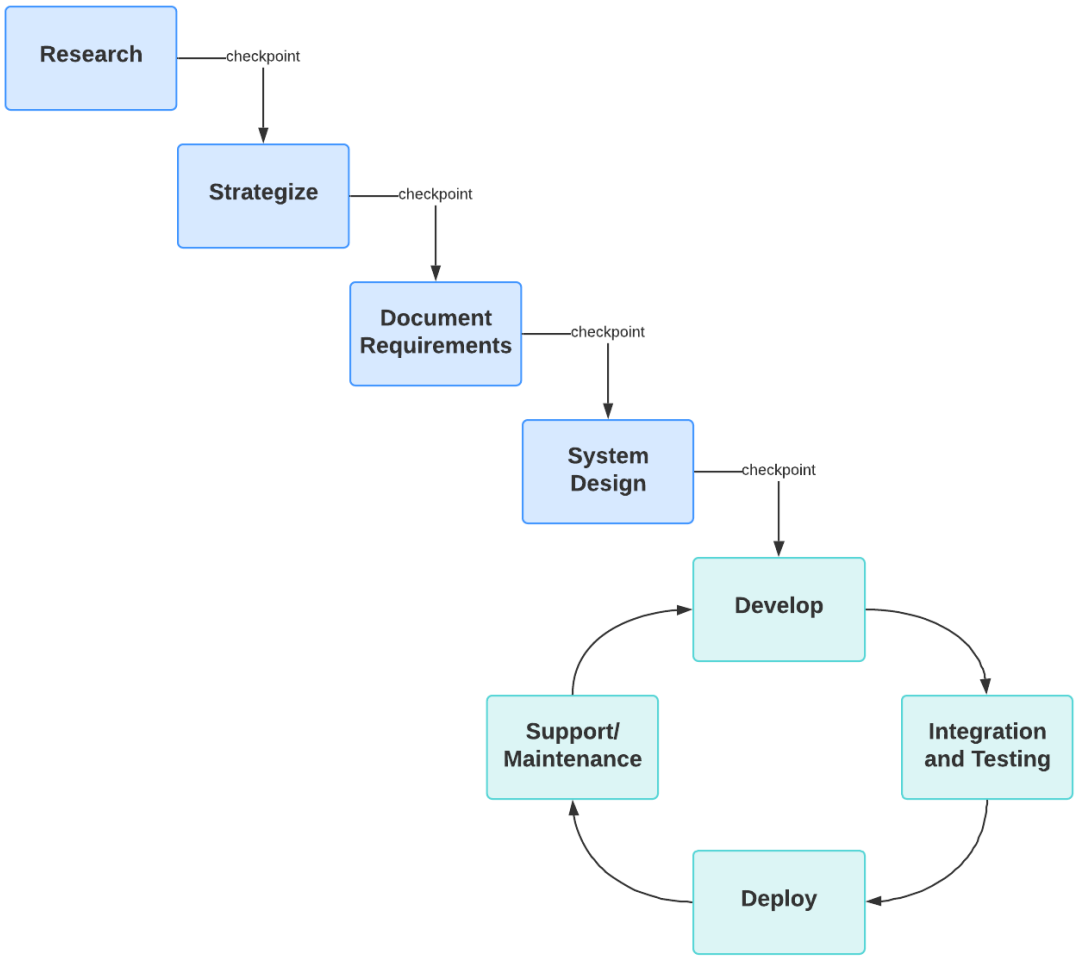
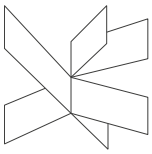


Figure 1: Agile-Waterfall Hybrid method (E. Bergmann & A. Hamilton, lucidchart.com)



This methodology has been chosen because the project has a delivery date, hence it would benefit from Agile's fast design, analysis, and planning. Using Waterfall at the organization level (starting from Research, Strategy, Requirements, Design), and using the Agile approach at a project level (circulating between Development, Testing, Deployment, Support/Maintenance) was found the best fit for this project.

However, since the stakeholders' wishes and requirements were changing more often, going back to **Requirements** through the stakeholders' **Feedback** was necessary. Hence, in our case, the used methodology loops through the following:

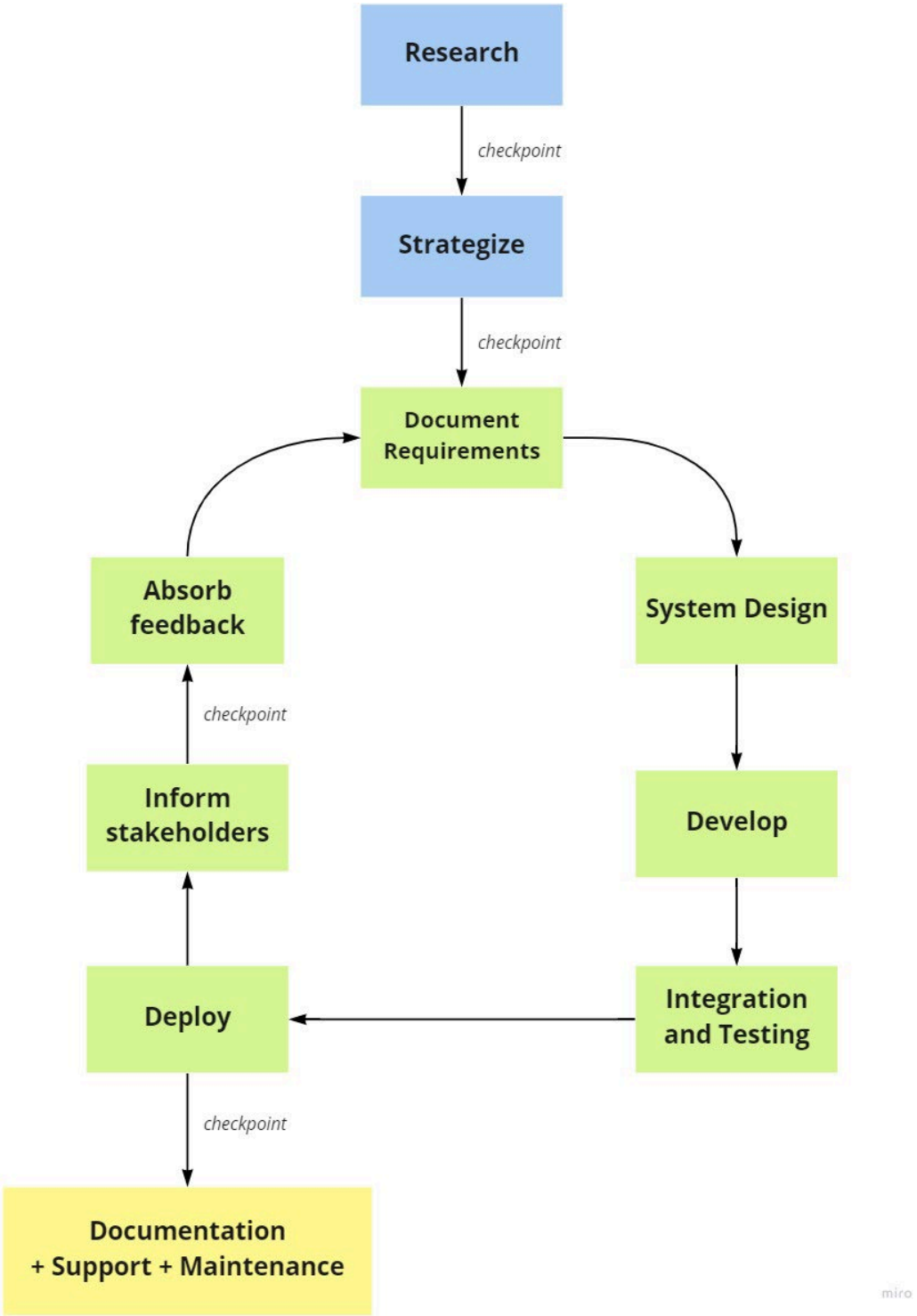
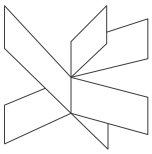


Figure 2: Agile-Waterfall Hybrid method – individual approach

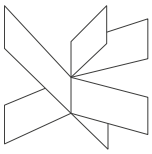


As can be seen on the previous figure, two extra phases have been added to our individual approach between **Deployment** and **Requirements**:

1. **Informing the stakeholders**

2. **Absorbing the stakeholders' Feedback**

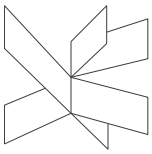
By doing so, the requirements can be updated accordingly.



6 Time schedule

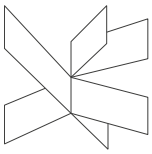
The Time schedule is based on CERN JIRA tickets, stakeholders' meetings and the need to have the site in production by the end of June 2022.
VIA deadlines are stated accordingly.

CERN Research	Early August 2021
CERN Mockups [Start]	05 August 2021
CERN Stakeholders' Meeting – Presentation of mockups	17 August 2021
CERN Mockups [End]	Early October 2021
CERN Strategy - Proposal for technology use	13 October 2021
CERN System Design - Architecture	Mid-October 2021
CERN Development - Implementation [Start]	14 October 2021
CERN Stakeholders' Meeting	14 October 2021
CERN Testing [Start]	1 November 2021
CERN Development - Implementation [End]	02 February 2022
CERN 1st Deployment	04 February 2022
CERN Documentation [Start]	11 February 2022
CERN Terra Incognita Presentation	28 February 2022
CERN Stakeholders' Meeting – Project Presentation	15 March 2022
VIA BPR1 Project description Draft [Start]	18 March 2022
VIA BPR1 Project description Draft [End]	31 March 2022
VIA BPR1 Project description Final [Start]	08 April 2022
VIA BPR1 Project description Final [End]	28 April 2022
VIA BPR1 Requirements/User Stories [Start]	29 April 2022
VIA BPR1 Requirements/User Stories [End]	09 May 2022



CERN Testing [End]	Mid-June 2022
CERN Documentation [End]	28 June 2022
CERN Project Delivery	30 June 2022
VIA BPR2 Final Report	March 2023

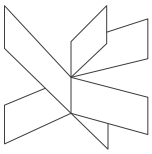
Figure 3: Time schedule



7 Risk assessment

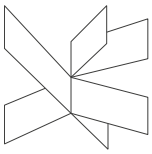
Risks	Likelihood Scale: 1-5 5=high risk	Severity Scale: 1-5 5=high risk	Product of likelihood and severity	Risk mitigation e.g Preventive-& Responsive actions	Identifiers	Responsible
Bad quality data, e.g.: a) corrupt video b) missing metadata c) typos in metadata that confuse the searches	2	4	8	Raise awareness with the sponsors	Bad searches	ATC chairperson
Non endorsement by the user community	3	3	9	Dissemination campaigns	Questions answered in the site	Site maintainer and the whole of the ATC
Software out of date or vulnerable for security	3	5	15	Maintenance processes well documented	Warnings in the repo	Site maintainer

Figure 4: Risk Assessment



8 Sources of Information

1. Bligh, D.A., 2000. Bligh, D. A. (2000). What's the use of lectures? San Francisco: Jossey-Bass. Jossey-Bass.
2. It-student-projects.web.cern.ch. 2022. CERN Academic Training web site | IT student projects. [online] Available at: <https://it-student-projects.web.cern.ch/projects/cern-academic-training-web-site> [Accessed 6 April 2022].
3. Youtube.com. 2022. CERN Lectures | YouTube. [online] Available at: <https://www.youtube.com/channel/UCwXkOx0EuKBR5m_OOiaZRUA/playlists> [Accessed 6 April 2022].
4. Cds.cern.ch. 2022. Academic Training Lectures - CERN Document Server. [online] Available at: <https://cds.cern.ch/collection/Academic%20Training%20Lectures?ln=en> [Accessed 6 April 2022].
5. Schools, S., Development, T. and Programme, A., 2022. Academic Training Lecture Regular Programme · Indico. [online] Indico. Available at: <https://indico.cern.ch/category/72/> [Accessed 6 April 2022].
6. Lucidchart.com. 2022. Agile-Waterfall Hybrid: Is It Right for Your Team? | Lucidchart Blog. [online] Available at: <https://www.lucidchart.com/blog/is-agile-waterfall-hybrid-right-for-your-team> [Accessed 6 April 2022].



Appendices

1. Appendix B – Group Contract

Group Contract

Group Name (optional): **Group 31** Date: **03/03/2022**
ATWEB

These are the terms of group conduct and cooperation that we agree on as a team.

Participation: We agree to....

- Be completely involved into the Academic Training web site project
- Be active during all the stages of the project
- Do our best to achieve the results defined in the project description

Communication: We agree to...

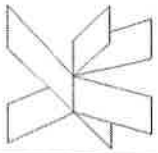
- Maintain a friendly and tolerant atmosphere in the group
- Listen to every group member's input / opinion
- Not ignore any of the members' messages

Meetings: We agree to....

- Have daily checkpoints between Franciska and Harris (can be postponed when necessary or omitted when there is nothing to discuss), other meetings scheduled together with other members when necessary
- The communication take place using Mattermost
- The meetings take place using Zoom (online) or using the office or social room (onsite) at CERN – depending on COVID restrictions
- Face-to-face and/or Mattermost discussions on the overall progress between Maria and Franciska on deadline, deliverables, interactions with the Academic Training Committee (ATC) and service managers of recording, transcoding and video publishing on the official CERN repository

Conduct: We agree to....

- Focus on discussed topic / tasks
- Track tasks with JIRA regularly



- Open SNOW Tickets at CERN Service Portal in case of other issues that cannot be solved on our own
- Keep backlog and other notes in CodiMD
- Regularly inform the BSc thesis supervisor on project progress (Maria, Kasper, Franciska)

Conflict: We agree to....

- In case a personal conflict occurs, we agree to respectfully express our thoughts and talk through them firstly in person.

Deadlines: We agree to....

- Do our best to meet the deadlines
- Warn the group members at least 3 days before deadline
- Franciska shares the drafts with Maria before handing in at VIA

Other Issues:

Technical Student	Student number (VIA)	Signature
Franciska-Leonóra Török (VIA)	293171	

Usability Tester	Signature
Salome Rohr (CERN)	S. ROHR

Project Maintainer	Signature
Harris Tzovanakis (CERN)	

Project Coordinator and Supervisor	Signature
Maria Dimou (CERN)	

Version: August, 2018
Template responsible: dans@via.dk

Source code

The source code is available on the following GitHub repository:
https://github.com/cern-sis/cern-academic-training

Source documentation

The source documentation is available on the following website:
https://academictraining-admin.docs.cern.ch/

BSc Thesis in Software Technology Engineering

User Guide

Building a website to promote the CERN Academic Training lectures

Franciska-Leonóra Török, 293171 IT

Supervisors:

Kasper Knop Rasmussen
VIA University College

Maria Dimou
CERN IT, Academic Training

7,698 characters

Software Technology Engineering

7th Semester

2023

Version: August, 2018
Template responsible: dans@via.dk

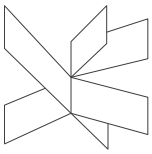
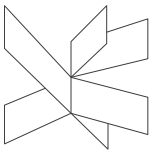


Table of content

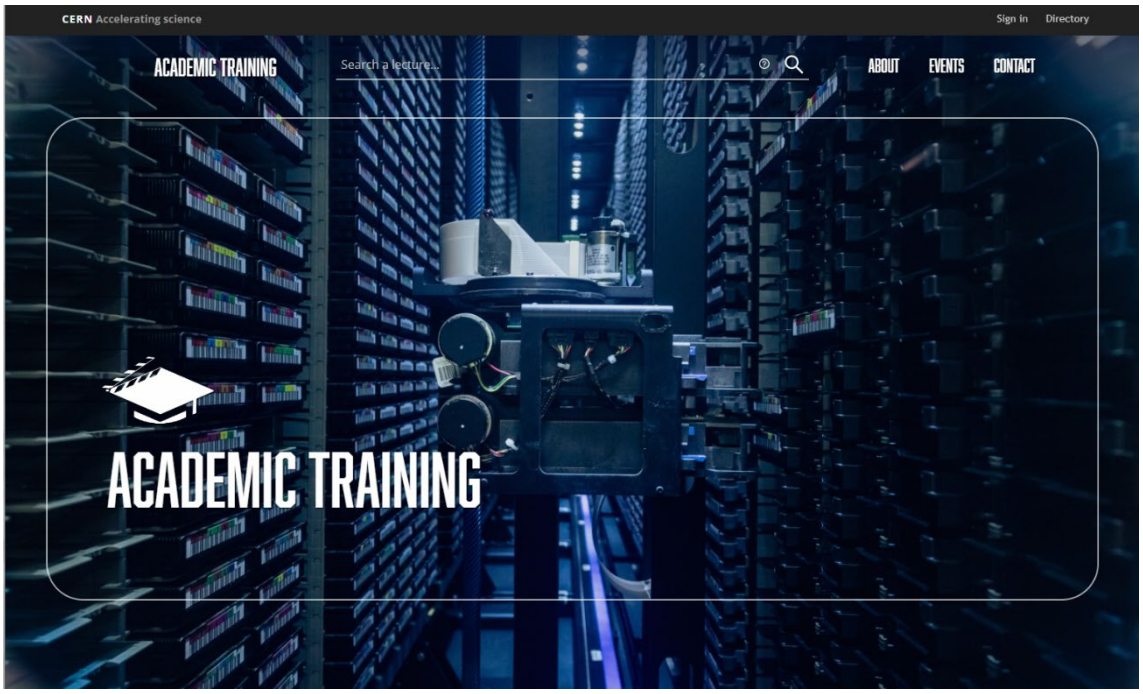


User Guide for Desktop

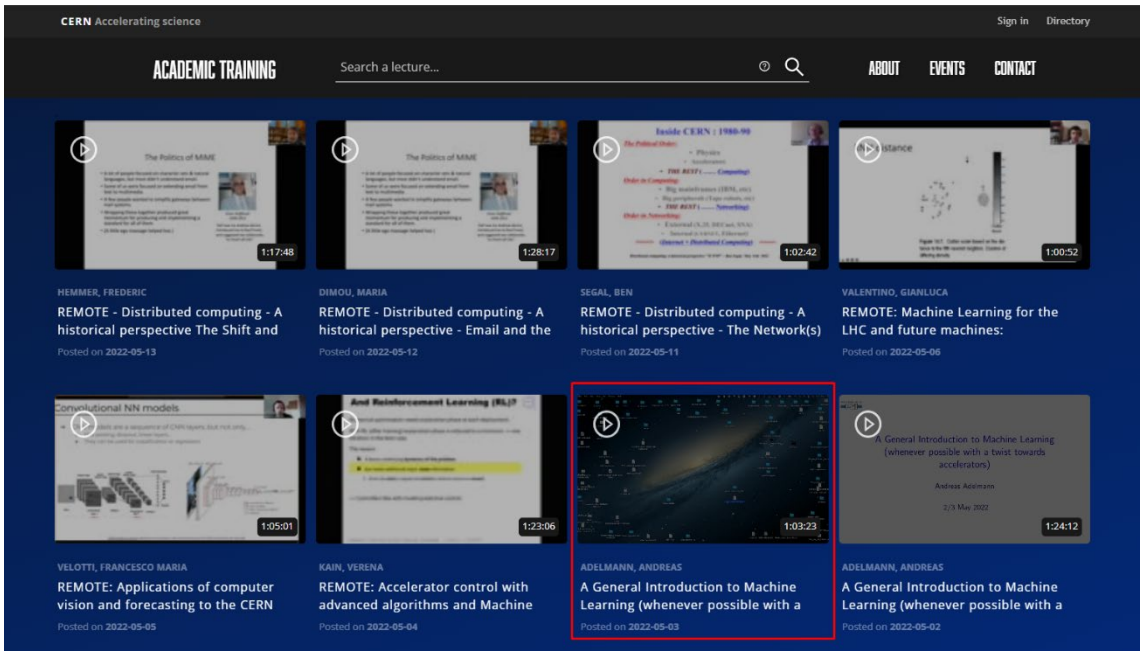
1 How to view a lecture?

There are two ways to view a lecture. One of them is directly accessing it from the list of most recently published lectures and the other way is by searching it through the search bar. The first option is going to be presented here.

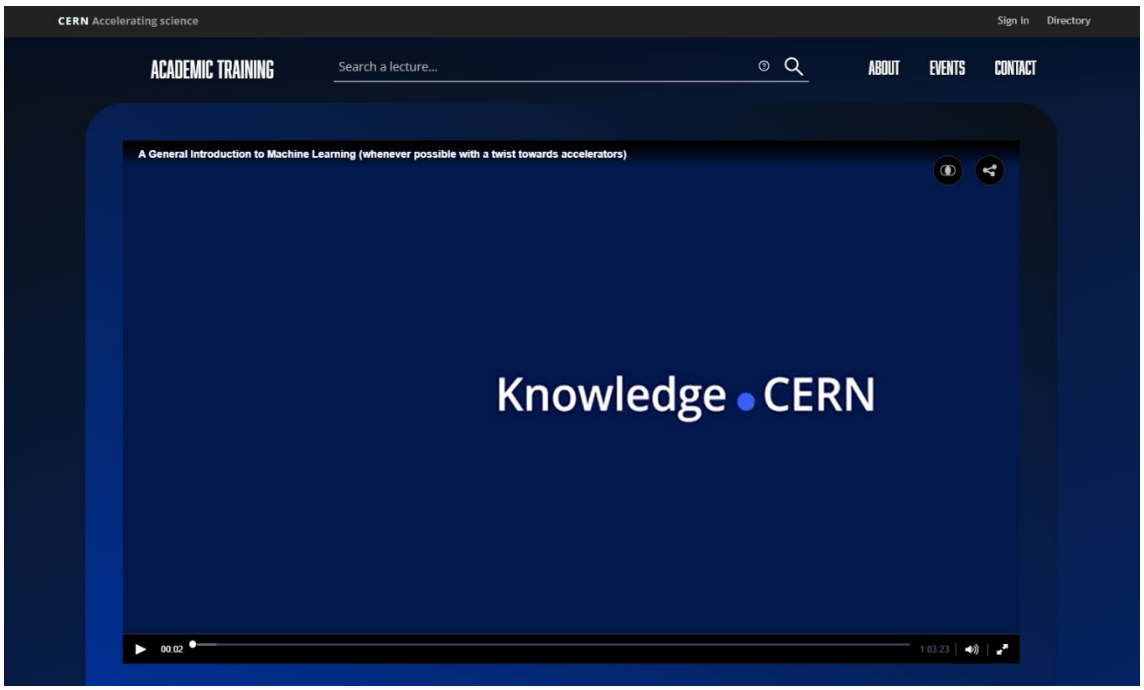
1. Go to the *Main Page* and scroll down.



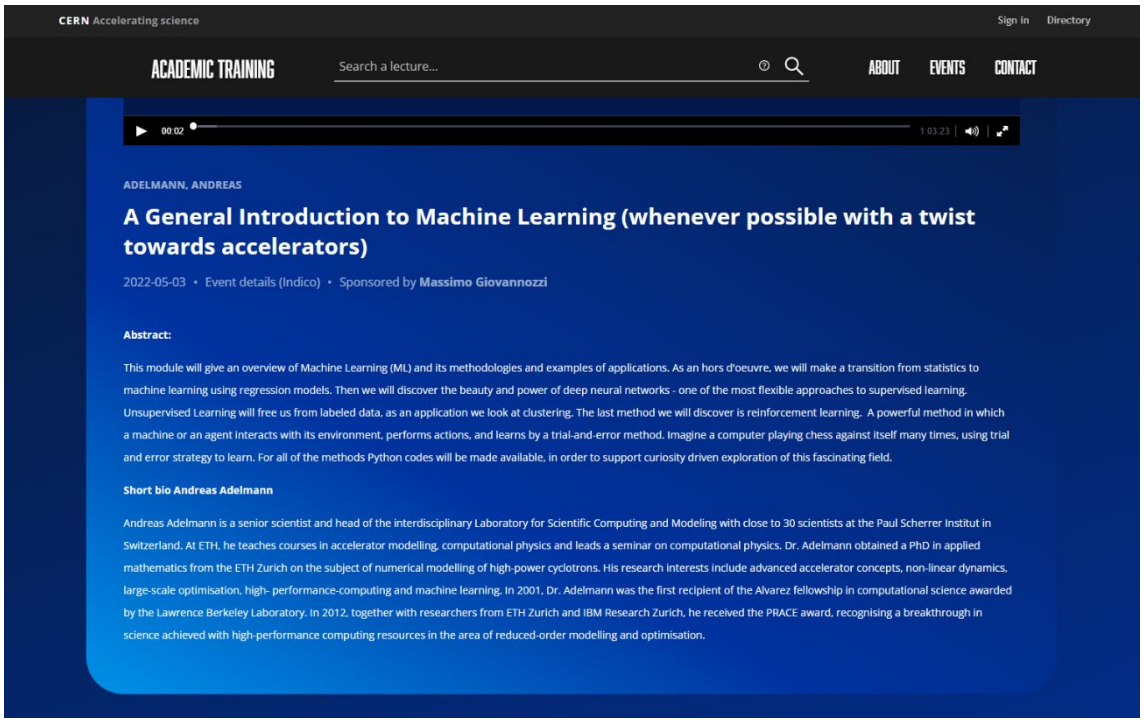
2. Find a lecture from the displayed list and click on it.



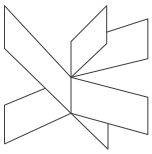
3. Click on the play button and watch the lecture.



4. Scroll down to find more details about the lecture.



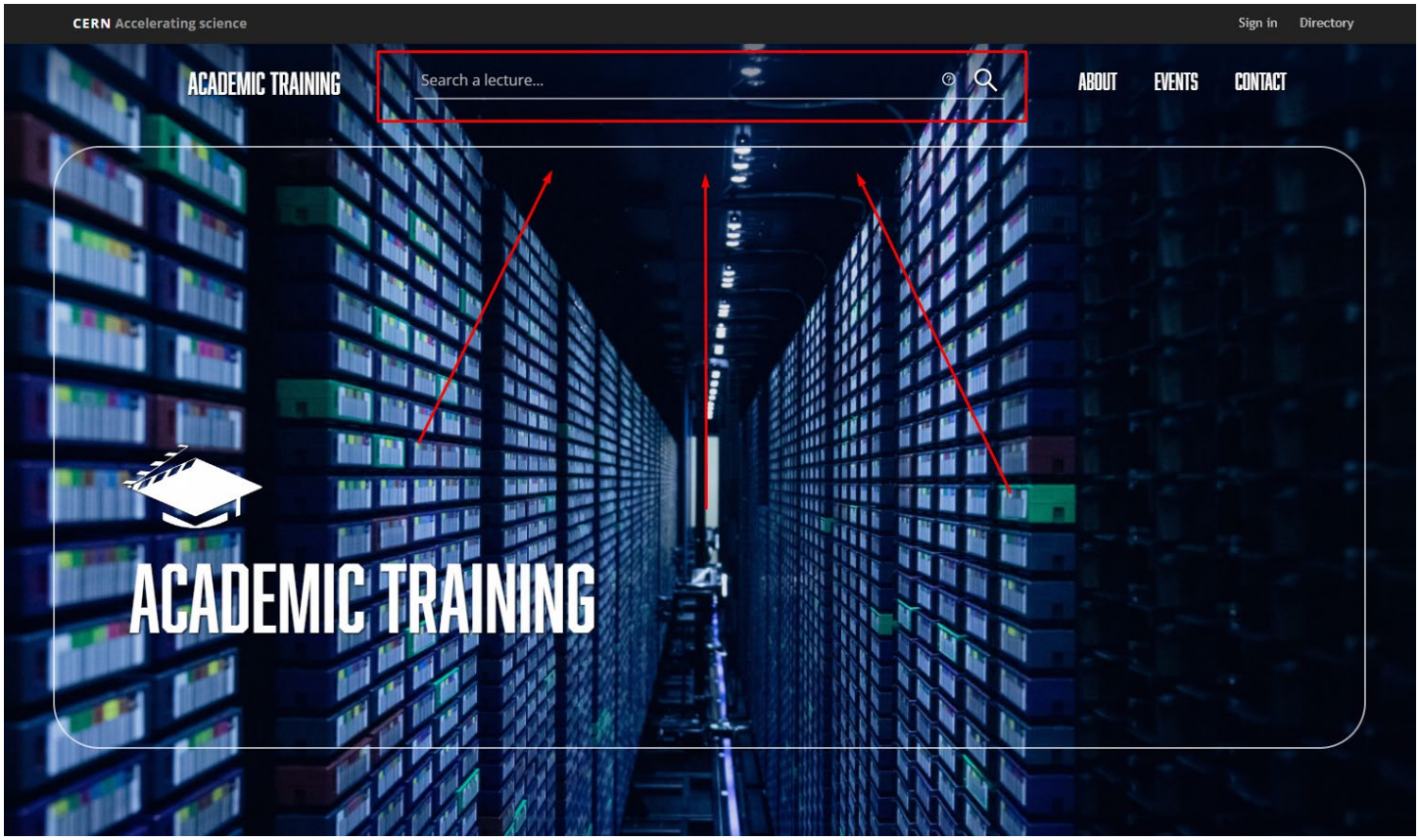
Note: Two-channel videos and non-video lectures work similarly. Examples for those are not shown here.



2 How to search a lecture?

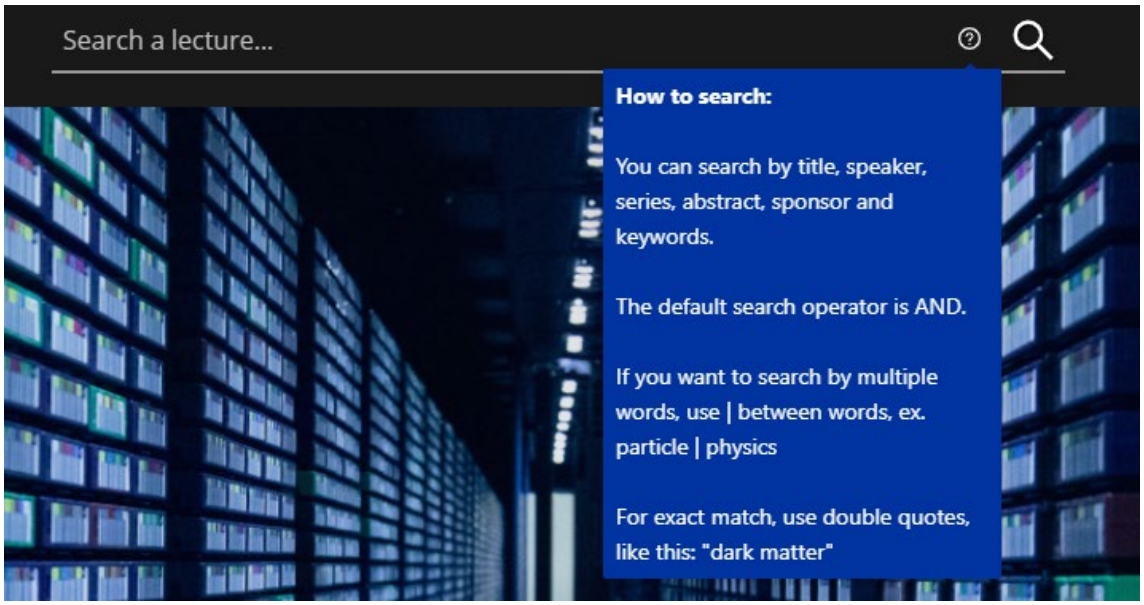
As mentioned earlier, viewing a lecture can be achieved by searching it too.

1. Find the search bar on the header of the site, right below the CERN toolbar.

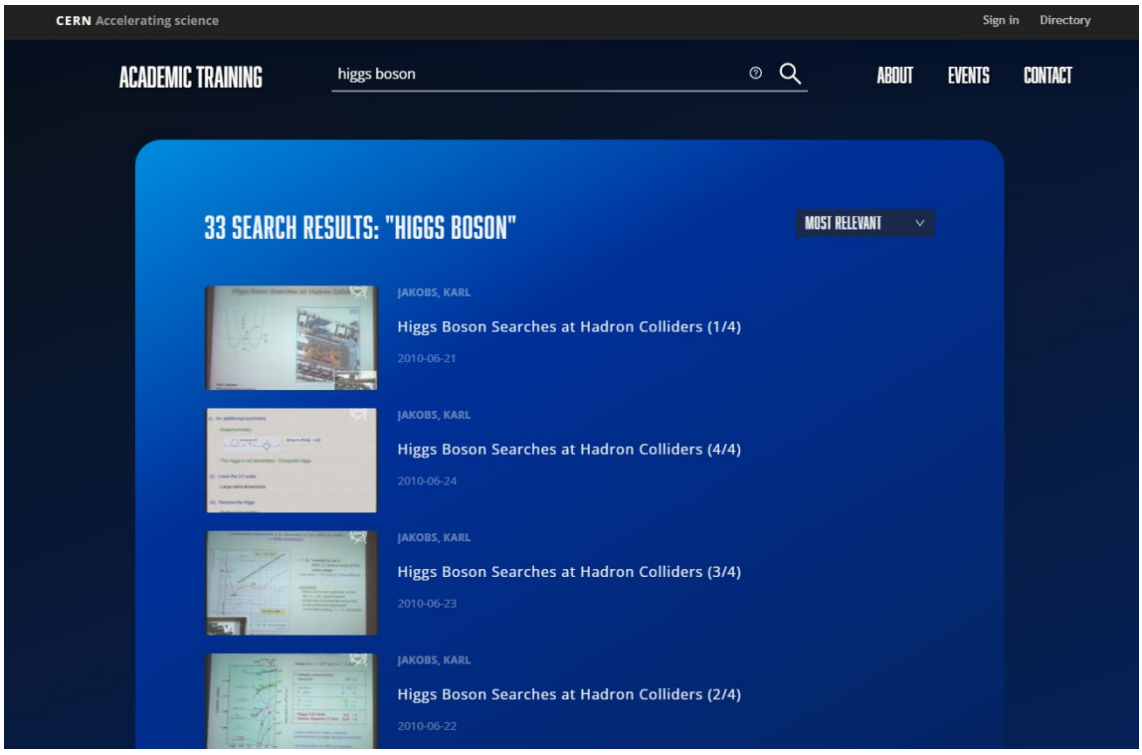


2. Write something in the search bar. You can search by title, abstract, speaker, date or sponsor. Use for example: *higgs boson*. For exact matches use double quotes, like *"higgs boson"*.

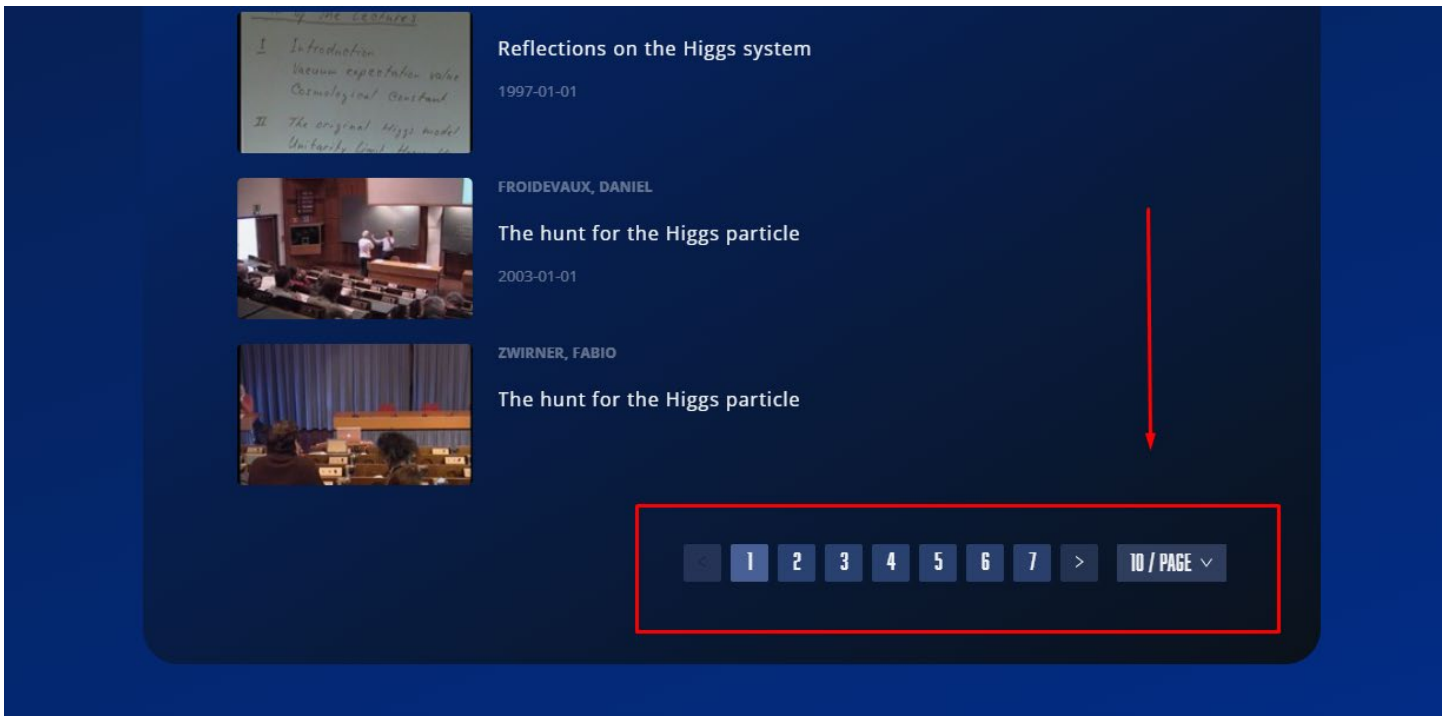
Note: You can find more help if you hover on the (?) tooltip next to the search button.



3. Find below the search results. Click on the lectures that you would like to check.

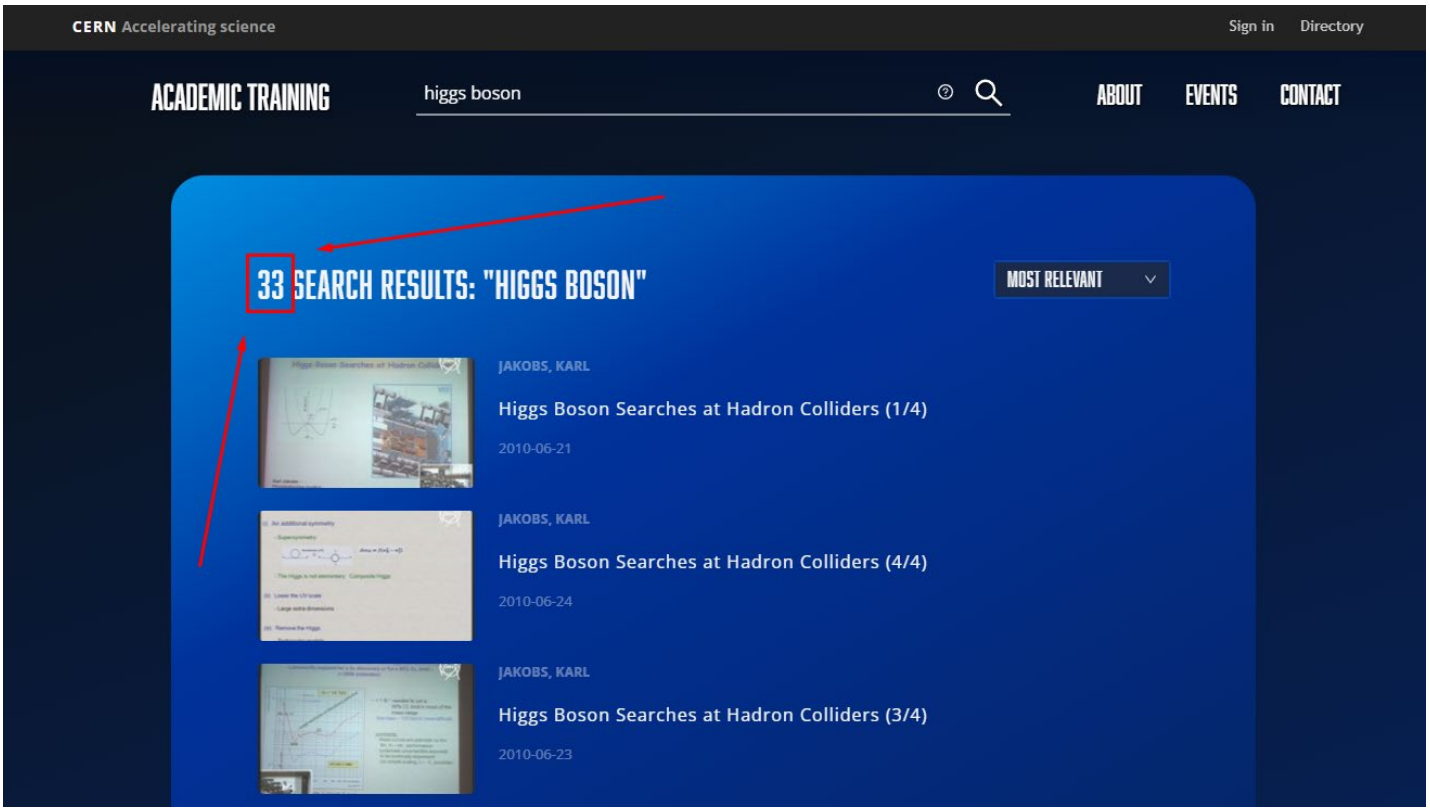


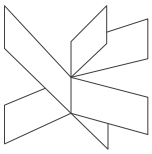
4. Scroll down for more results. Use the pagination to find even more lectures related to the searched term.



3 How to find the number of results from search?

The number of results can be found at the top-left corner of the *Search Page*.

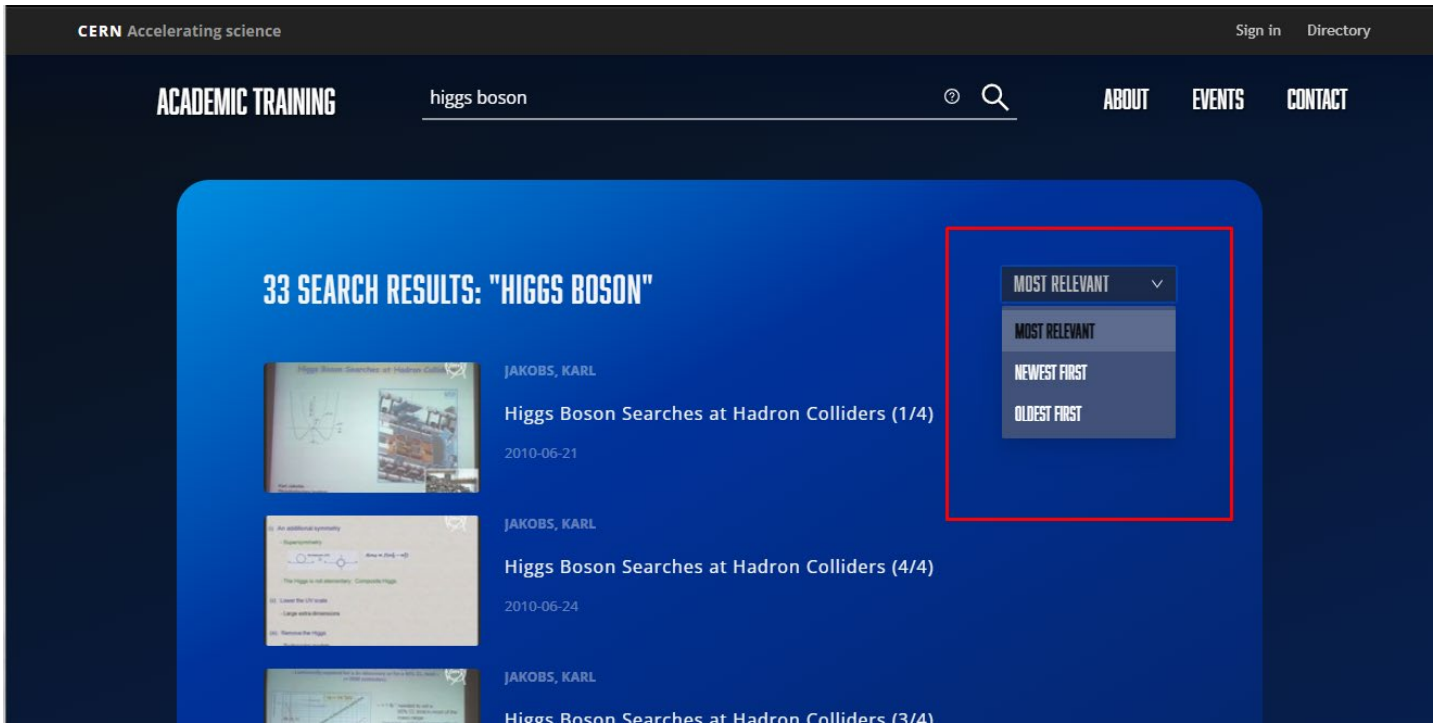




4 How to sort the search results by relevance?

If you would like to sort the lectures by relevance, you must click on the dropdown menu at the top-right corner of the *Search Page*. The following is going to be listed:

1. Most relevant
2. Newest first
3. Oldest first



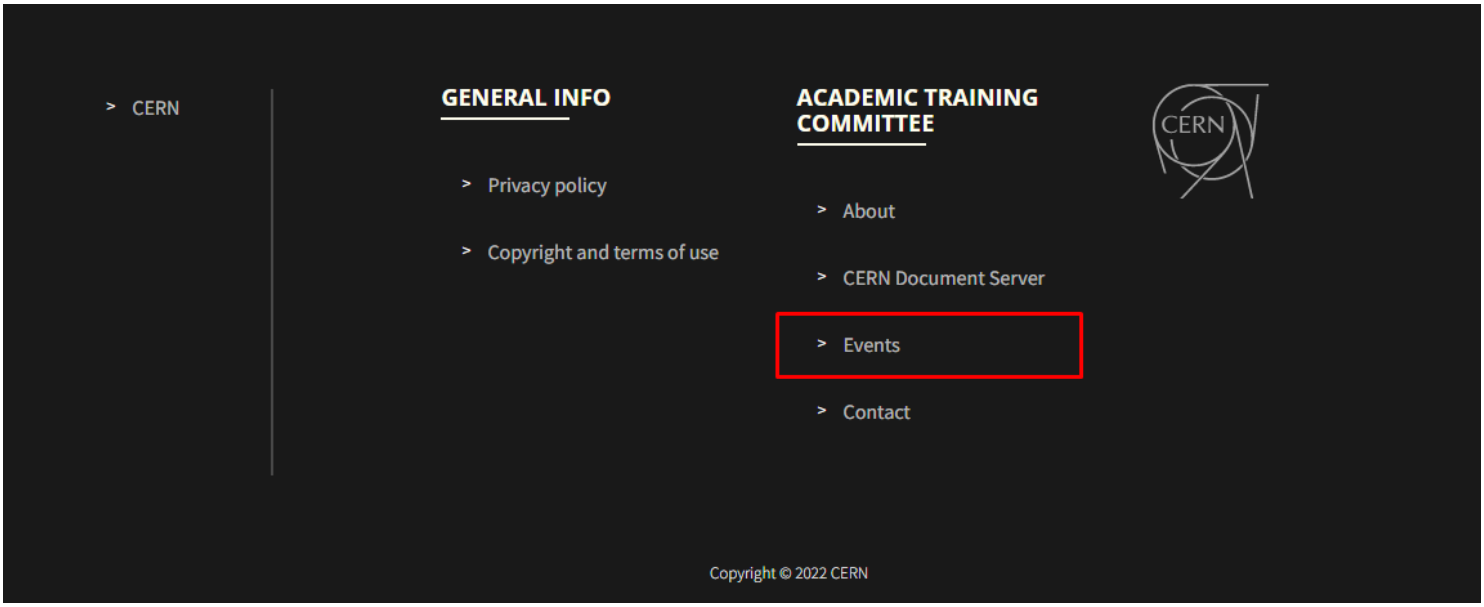
5 How to access the full agenda of the Academic Training lectures?

There are two ways to find the full programme of the Academic Training lectures.

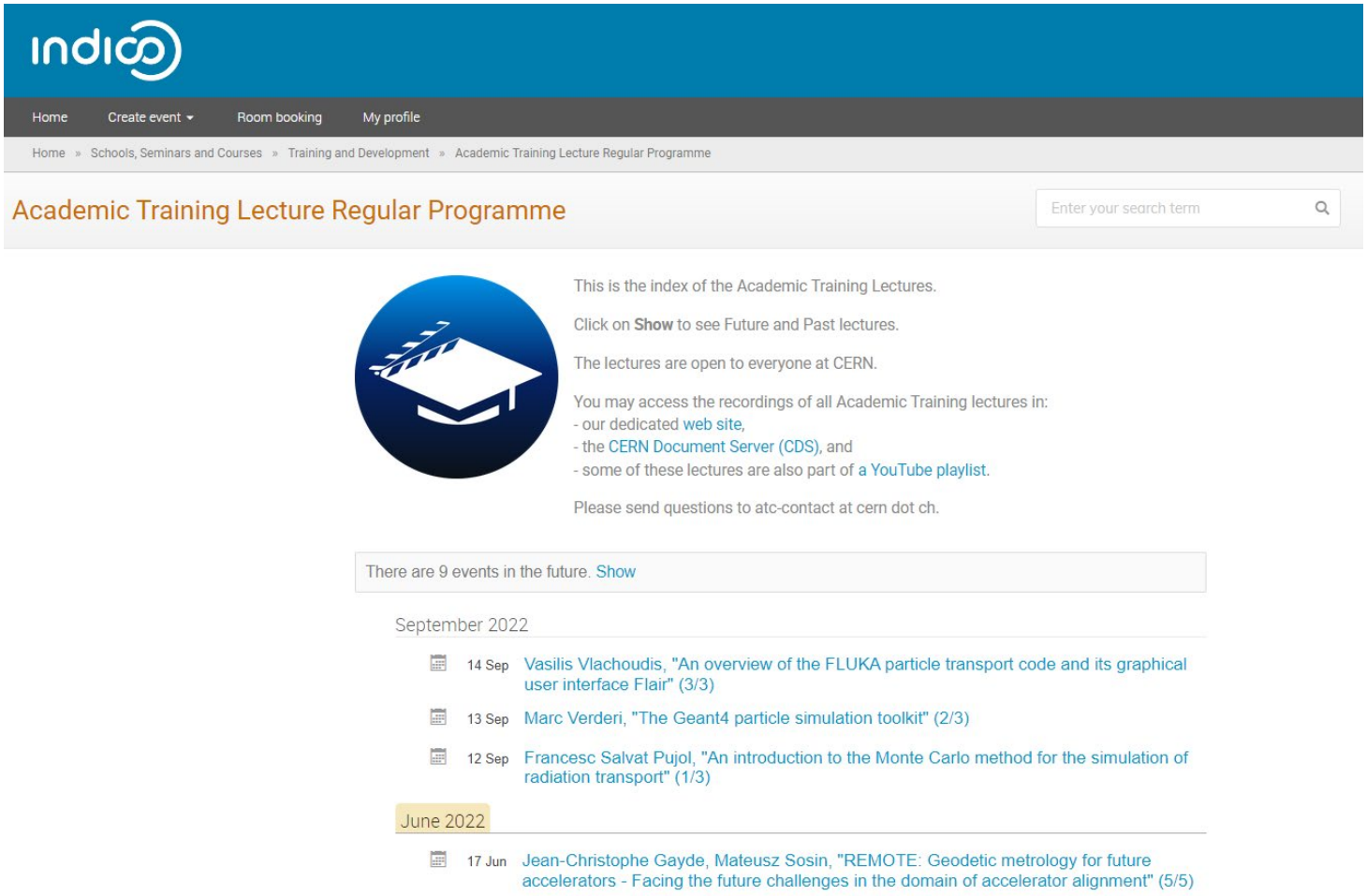
1. Go to the header and find the **EVENTS** button. Click on it.

OR

2. Scroll down to the footer and find the **Events** button under the Academic Training Committee block. Click on it.



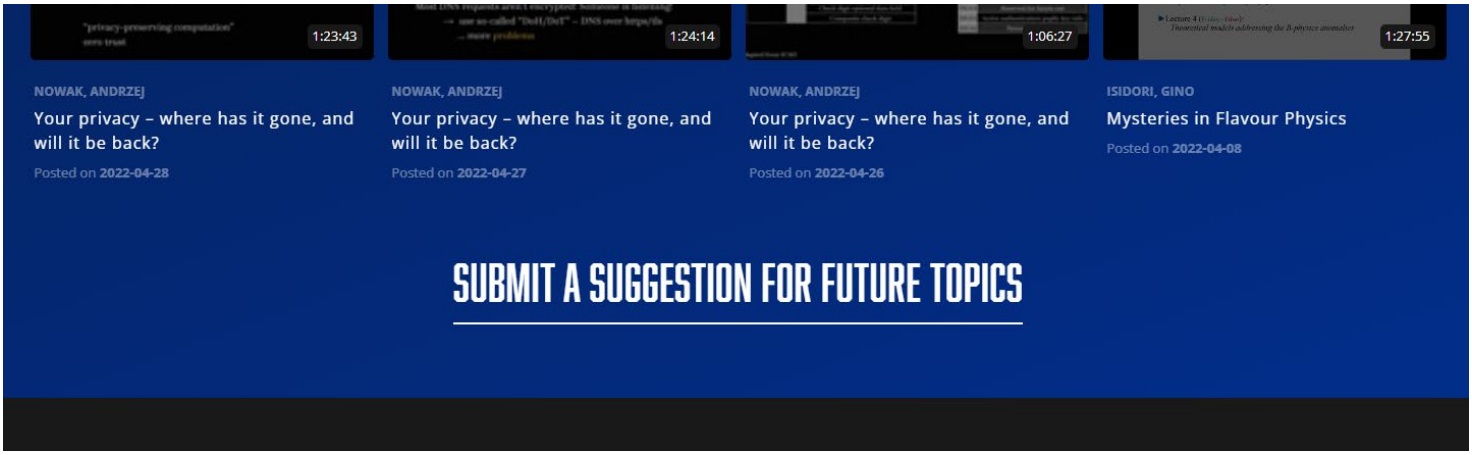
It should navigate you to the Indico page of the full *Academic Training Lecture Regular Programme*.



6 How to contact the ATC?

There are three ways of contacting the ATC (*Academic Training Committee)*.

1. Go to the *Main Page*, scroll down and find the **SUBMIT A SUGGESTION FOR FUTURE TOPICS** button. Click on it.

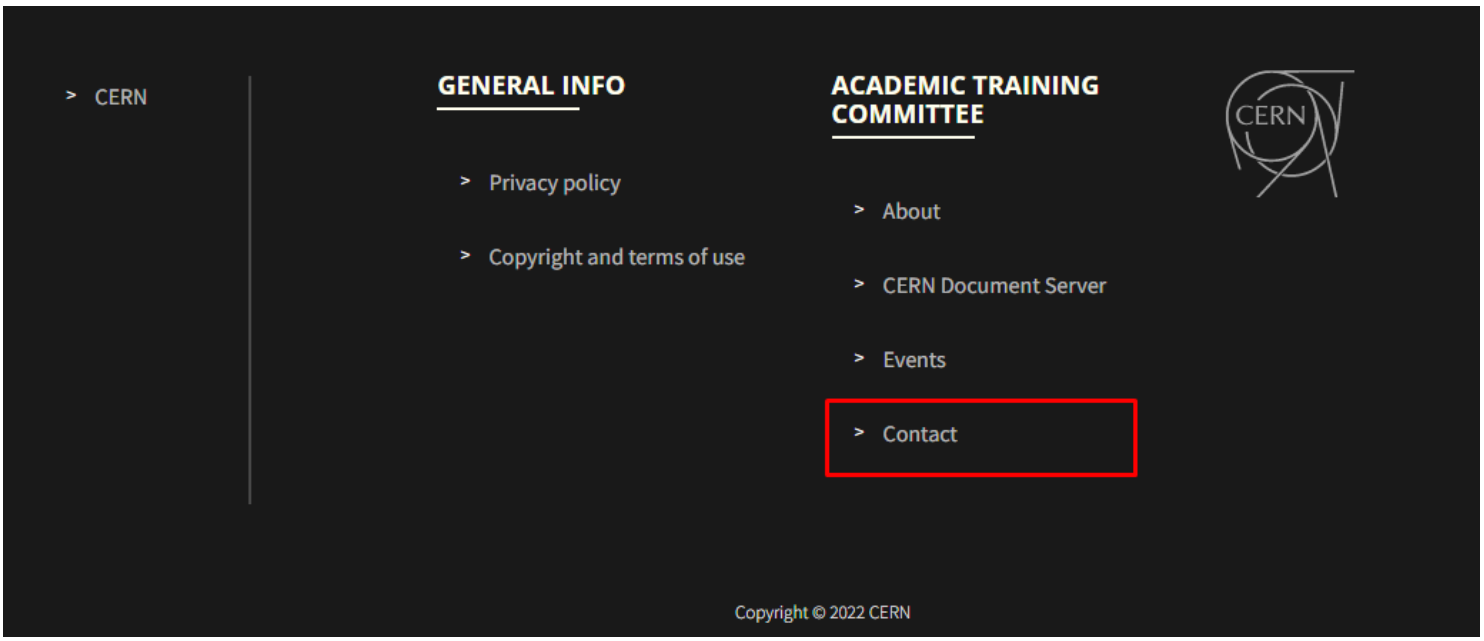


OR

2. Go to the header and find the **CONTACT** button. Click on it.

OR

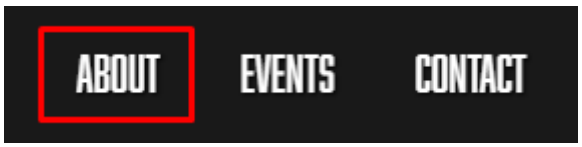
3. Scroll down to the footer and find the **Contact** button under the Academic Training Committee block. Click on it.



All of the above shall open your default email agent with the address *atc-contact@cern.ch*.

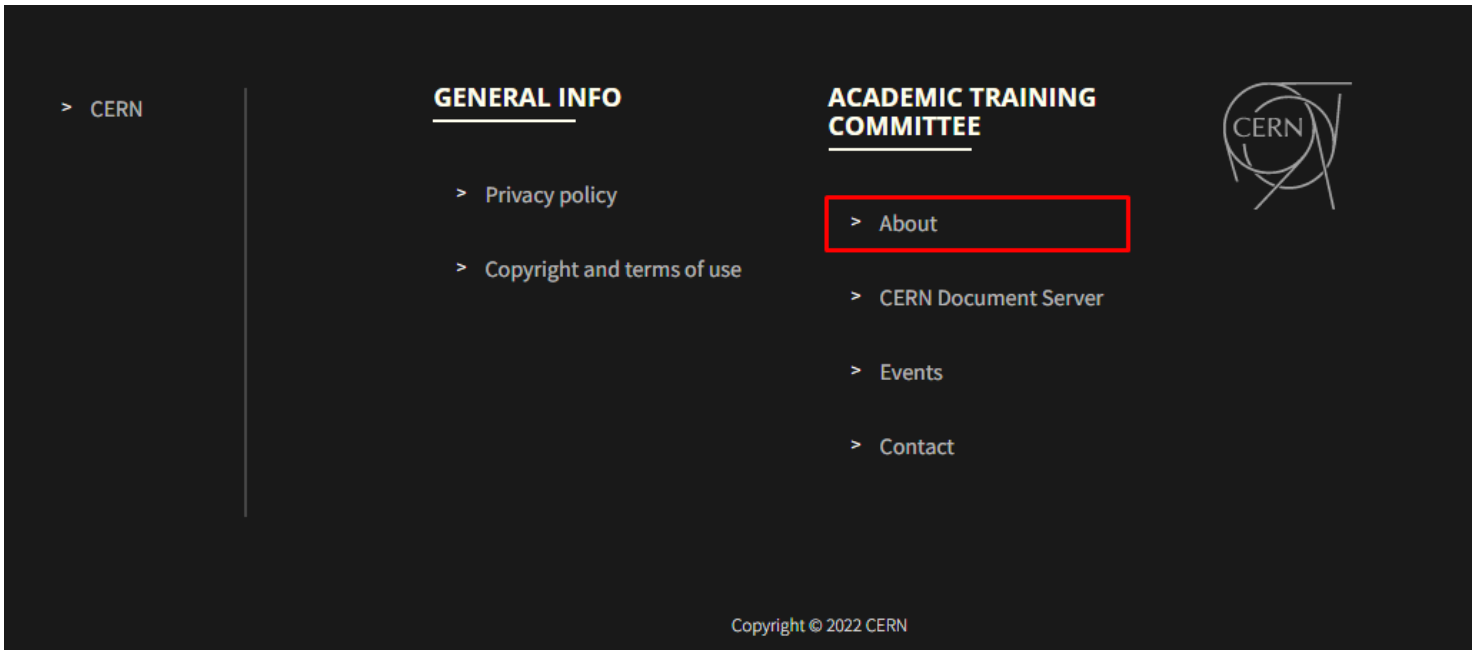
7 How to view who the members of the ATC are?

1. Go to the header and find the ABOUT button. Click on it.

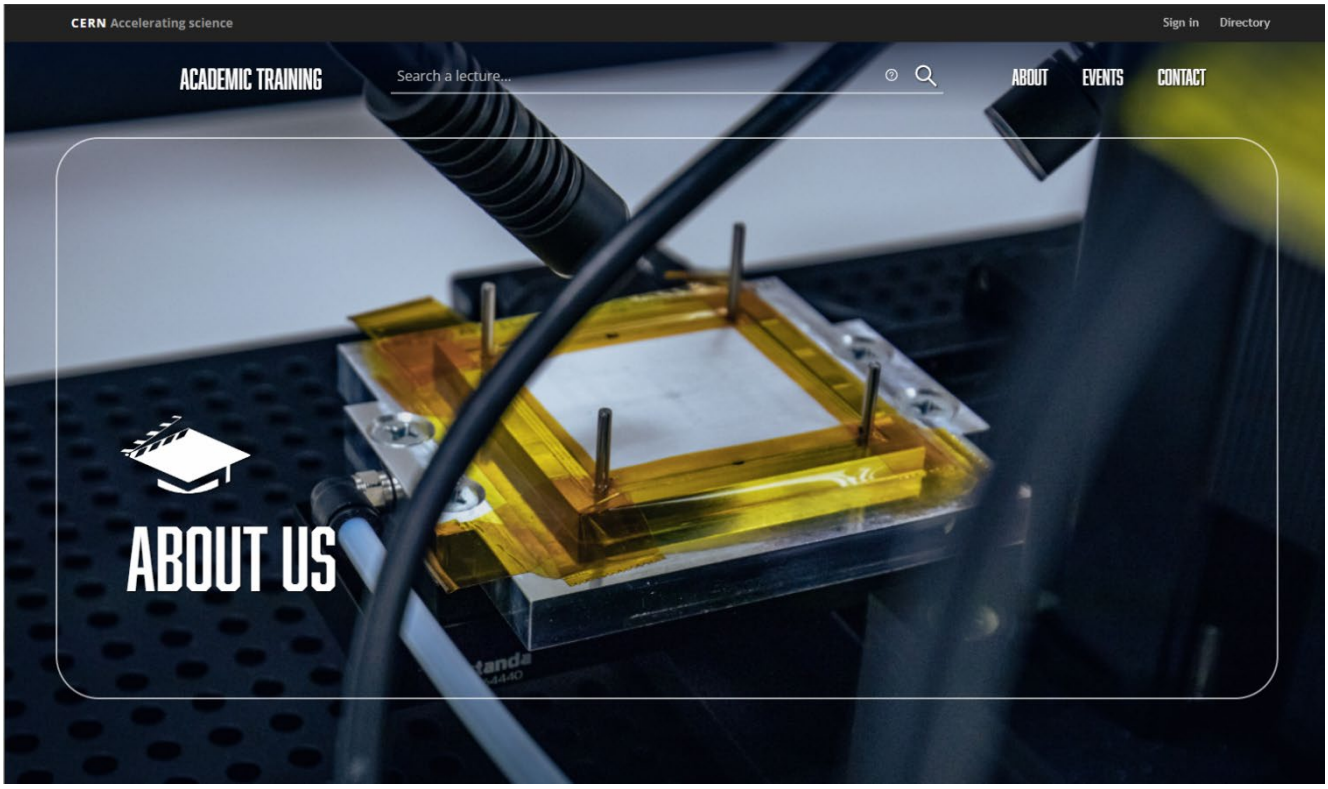


OR

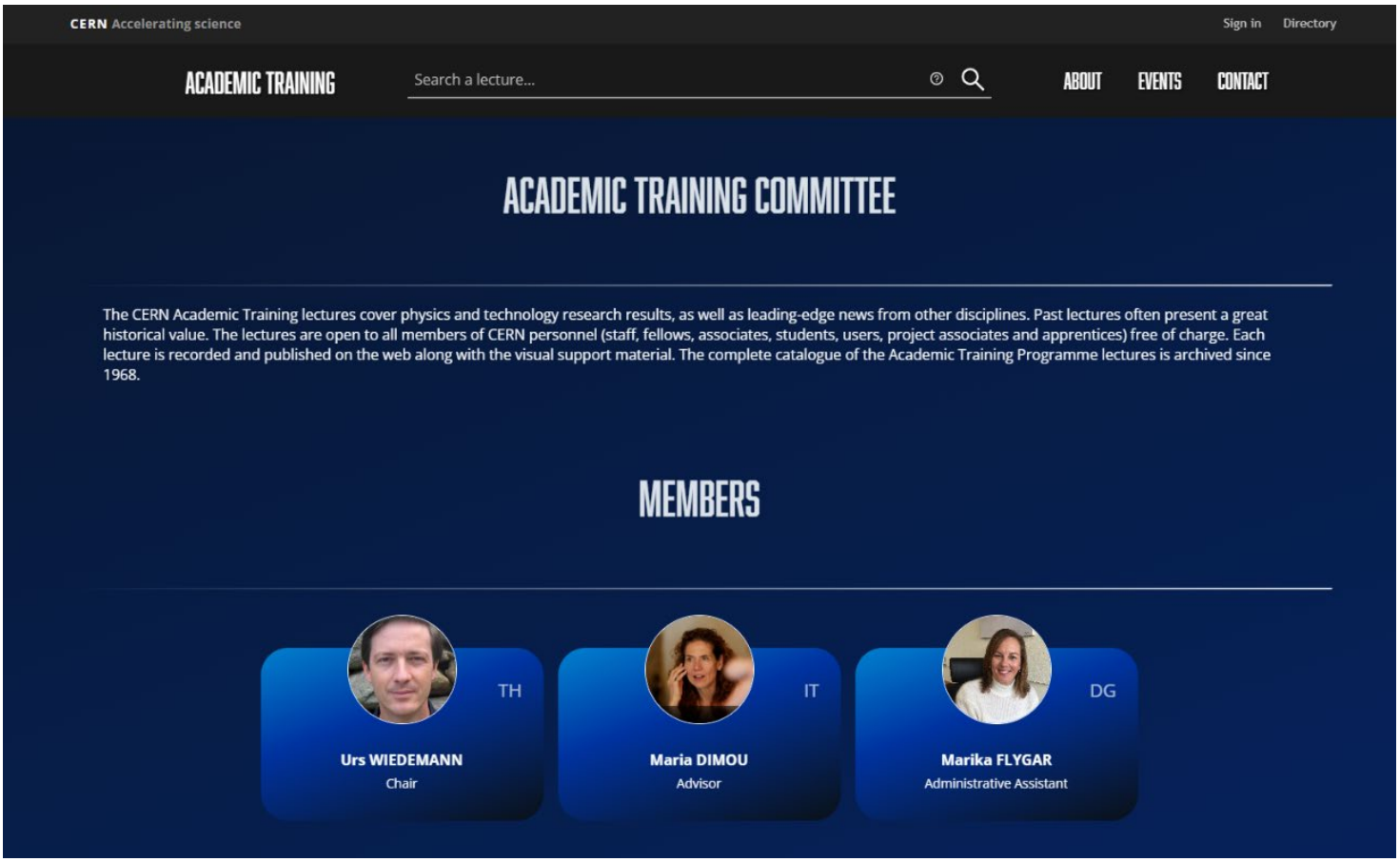
2. Scroll down to the footer and find the About button under the Academic Training Committee block. Click on it.



All of the above shall open the *About Us Page* that displays the current members of the Academic Training Committee.

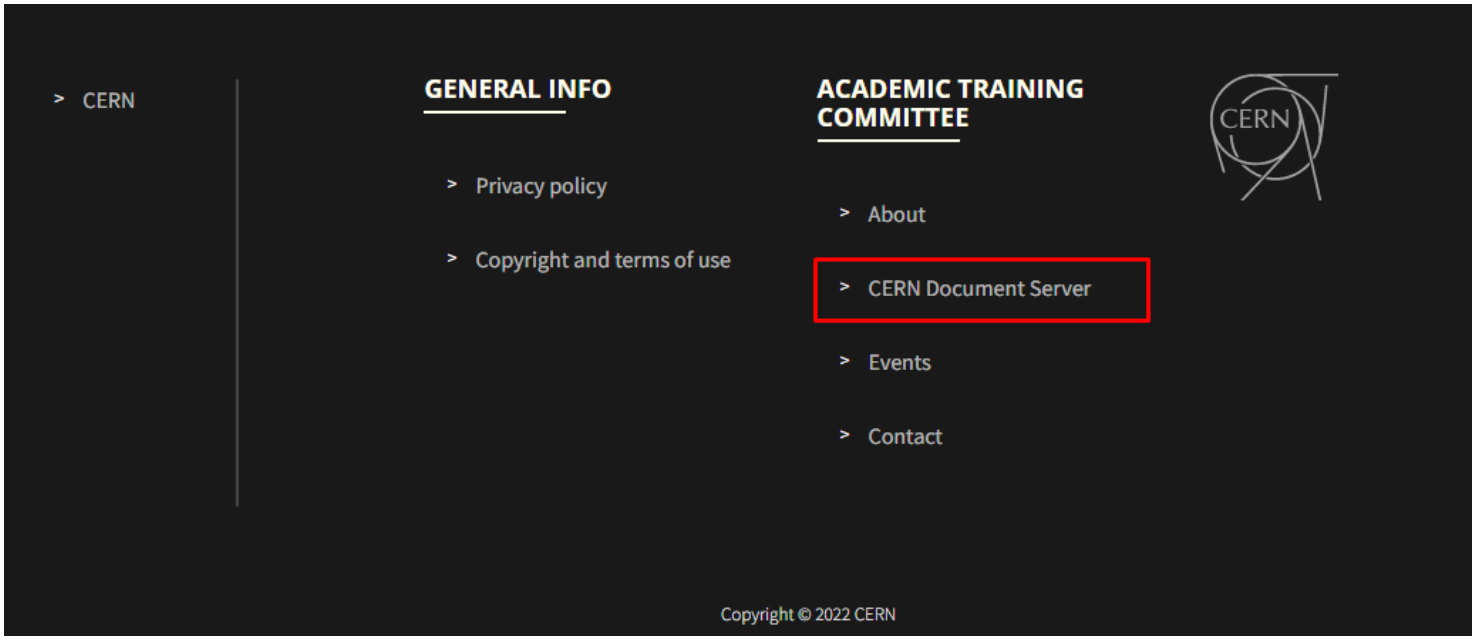


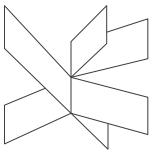
Scroll down to see the full list.



8 How to view the lectures from the CERN Document Server?

Scroll down to the footer and find the **CERN Document Server** button under the Academic Training Committee block. Click on it.



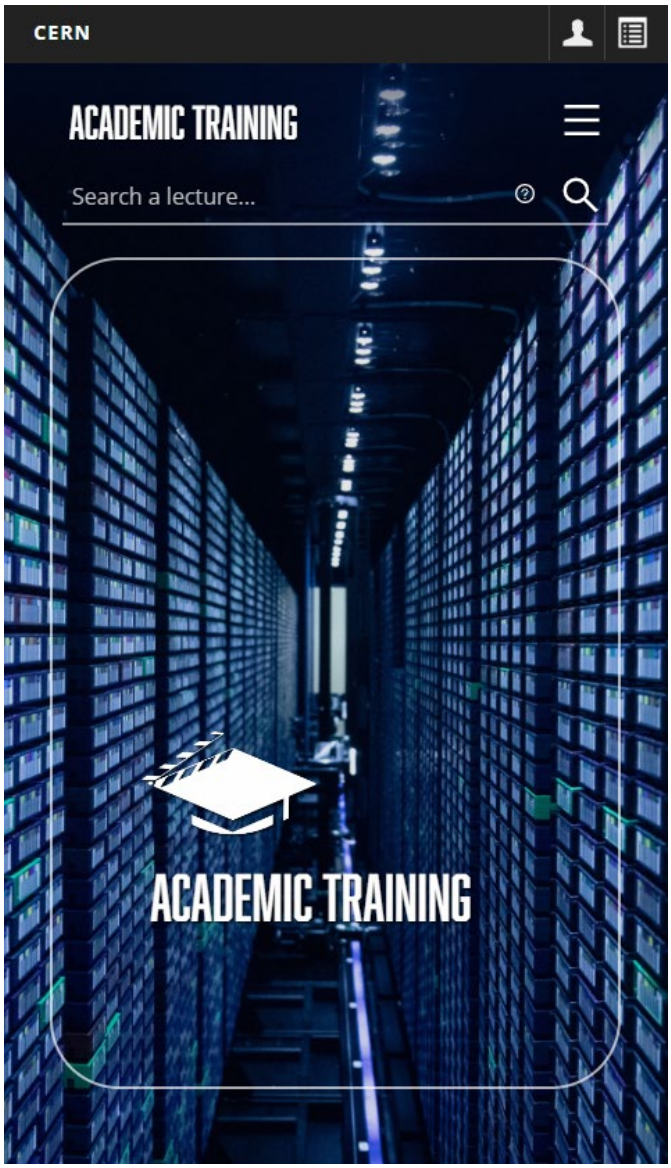


User Guide for Mobile

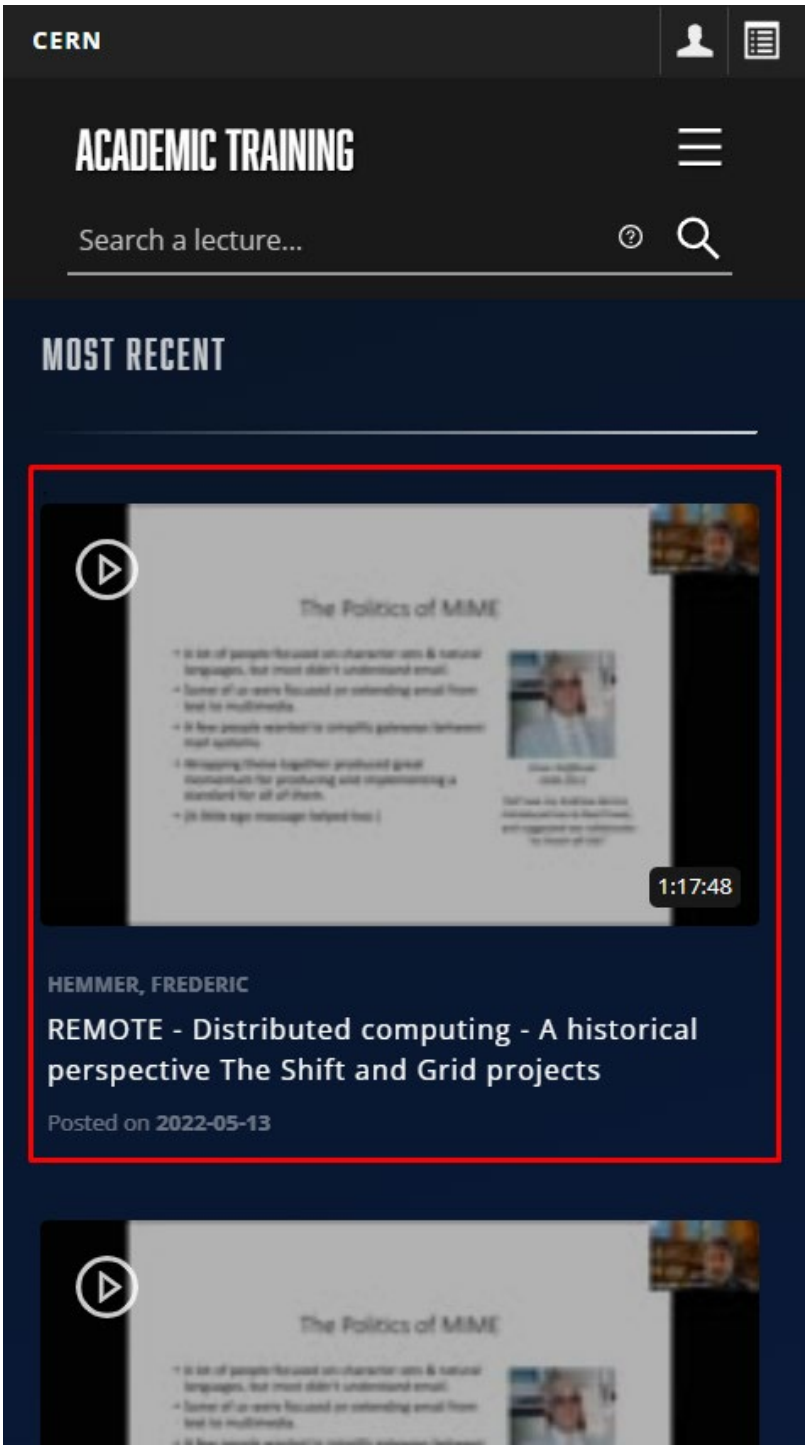
9 How to view a lecture?

There are two ways to view a lecture. One of them is directly accessing it from the list of most recently published lectures and the other way is by searching it through the search bar. The first option is going to be presented here.

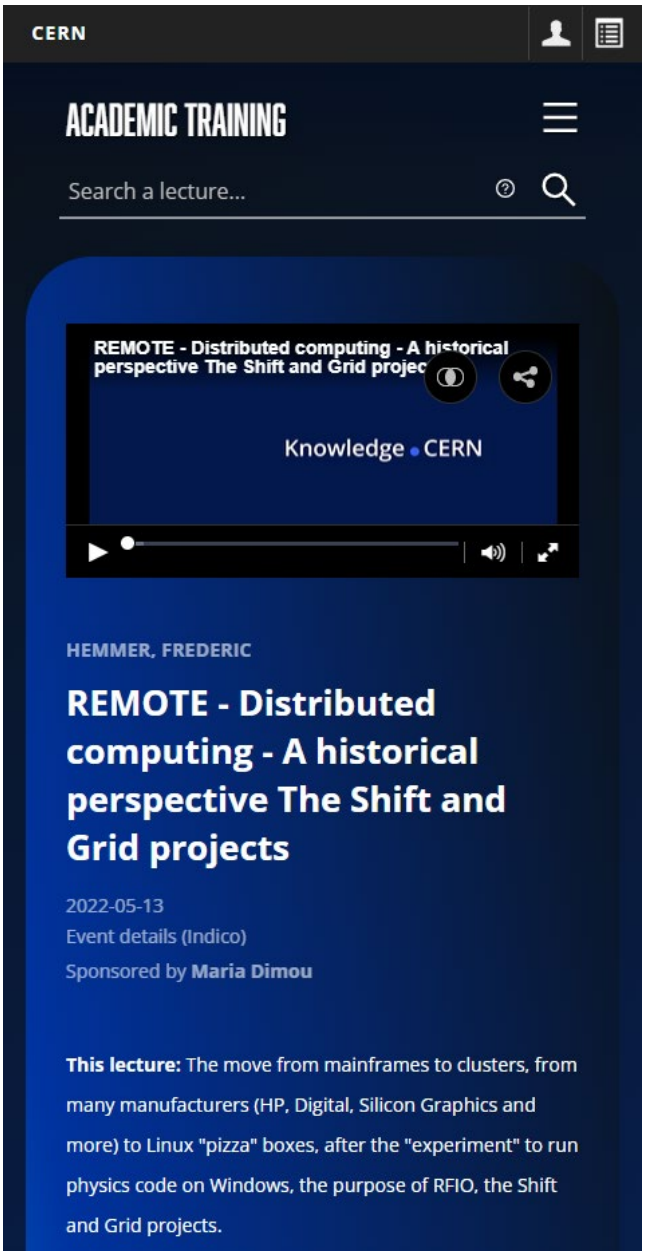
1. Go to the *Main Page* and scroll down.



2. Find a lecture from the displayed list and click on it.



3. Click on the play button and watch the lecture.



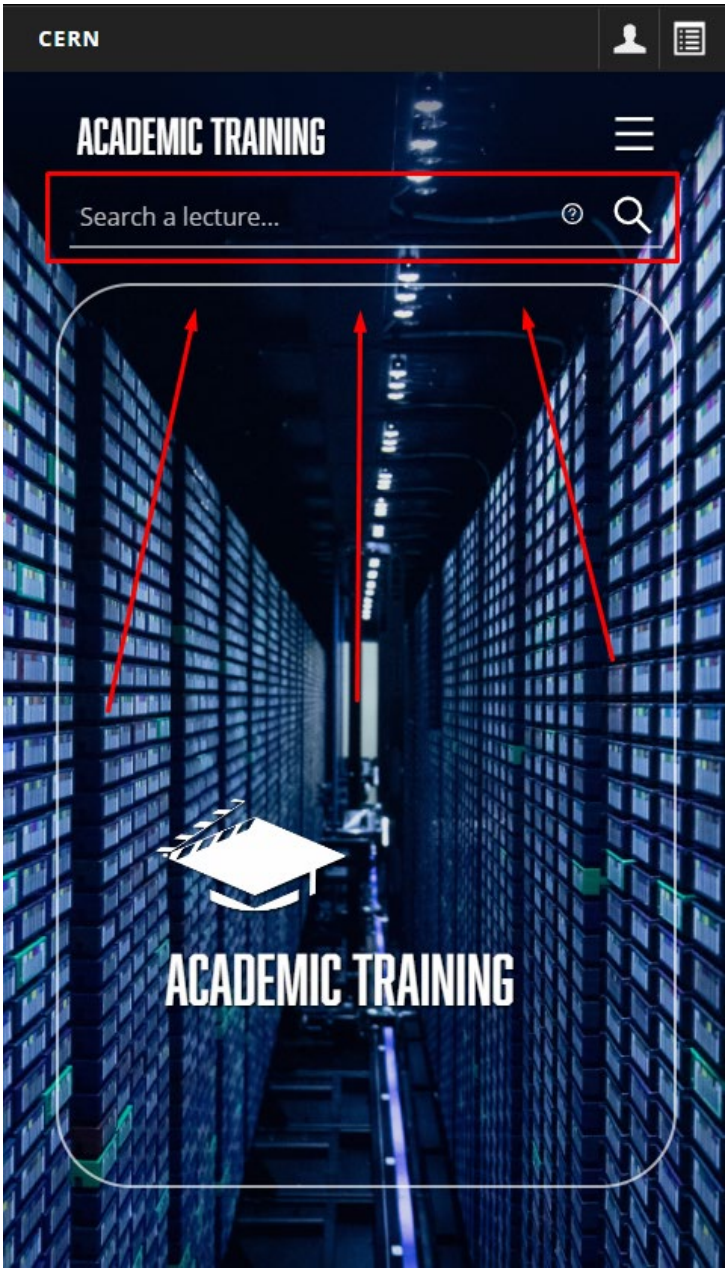
4. Scroll down to find more details about the lecture.

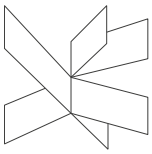
Note: Two-channel videos and non-video lectures work similarly. Examples for those are not shown here.

10 How to search a lecture?

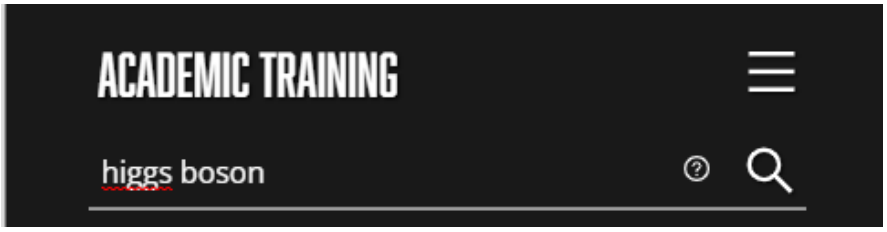
As mentioned earlier, viewing a lecture can be achieved by searching it too.

1. Find the search bar on the header of the site, right below the CERN toolbar.

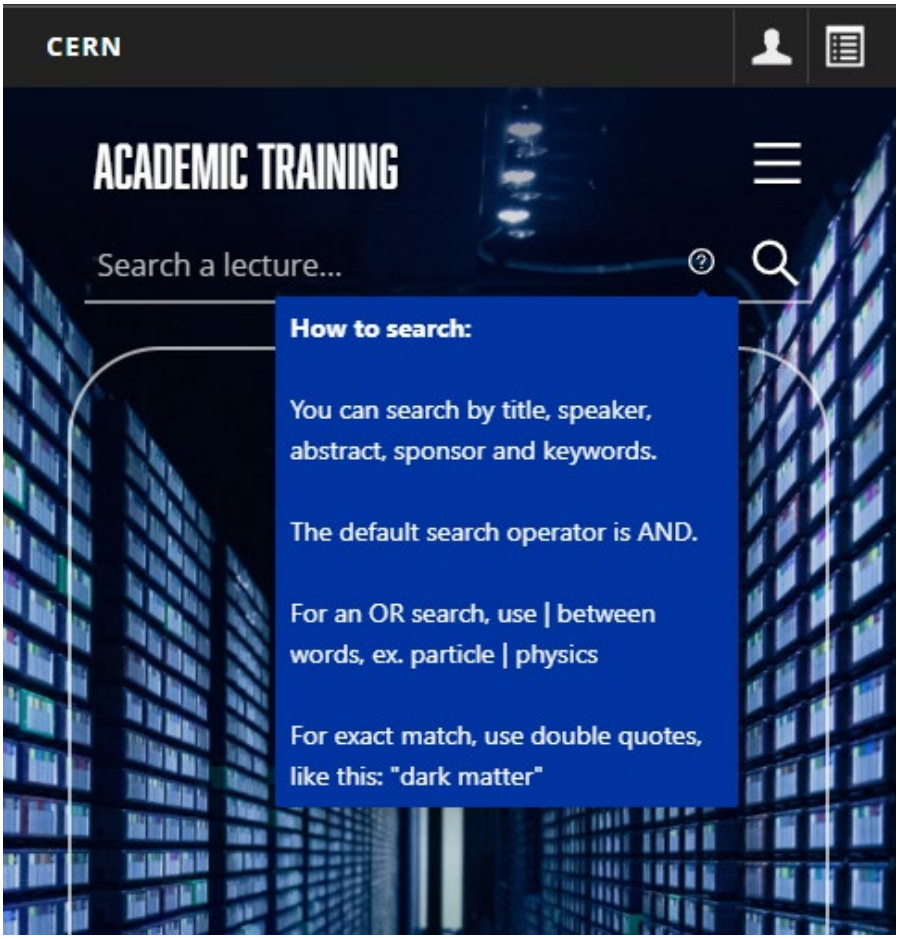




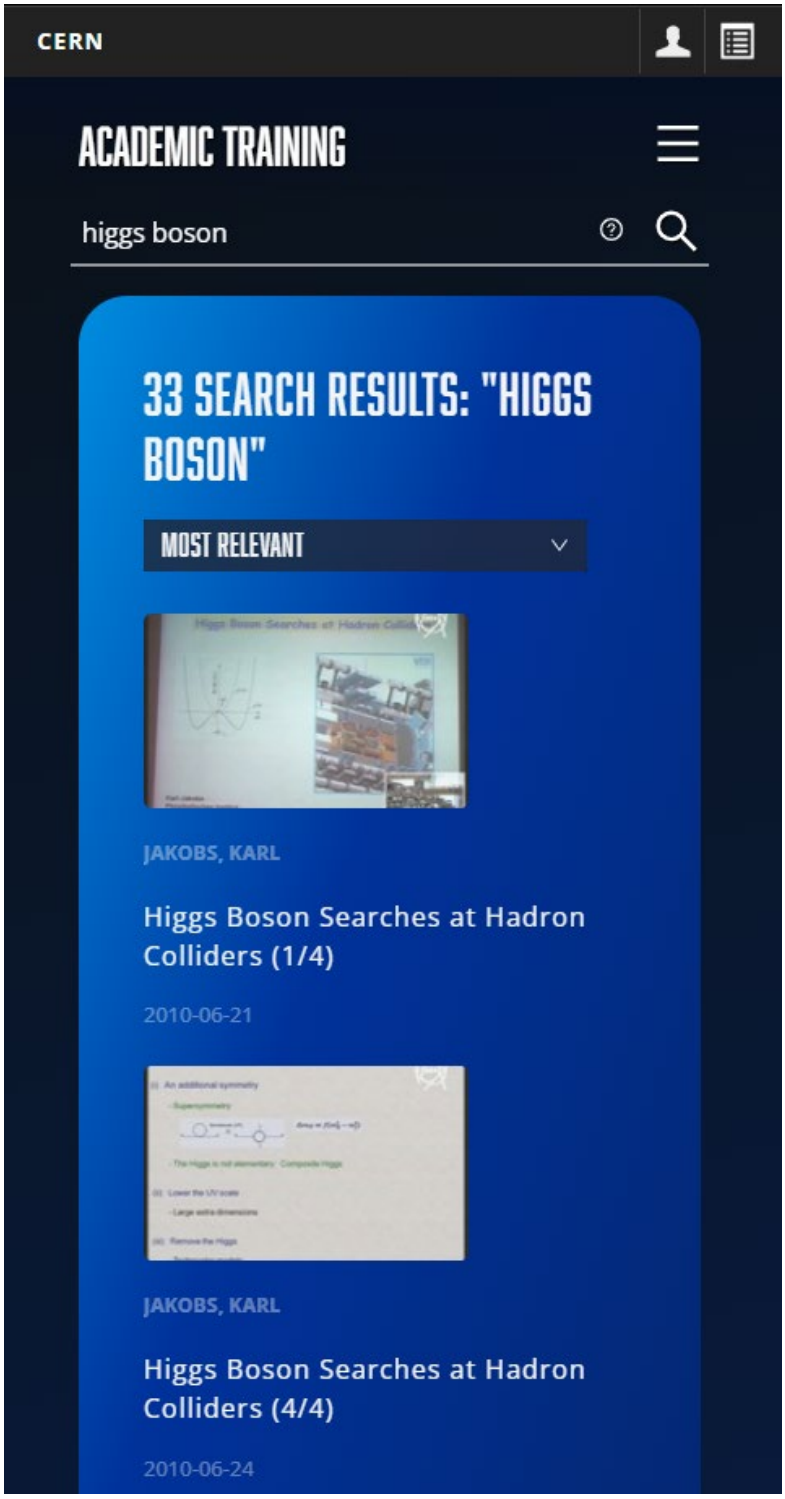
2. Write something in the search bar. You can search by title, abstract, speaker, date or sponsor. Use for example: *higgs boson*. For exact matches use double quotes, like *"higgs boson"*.



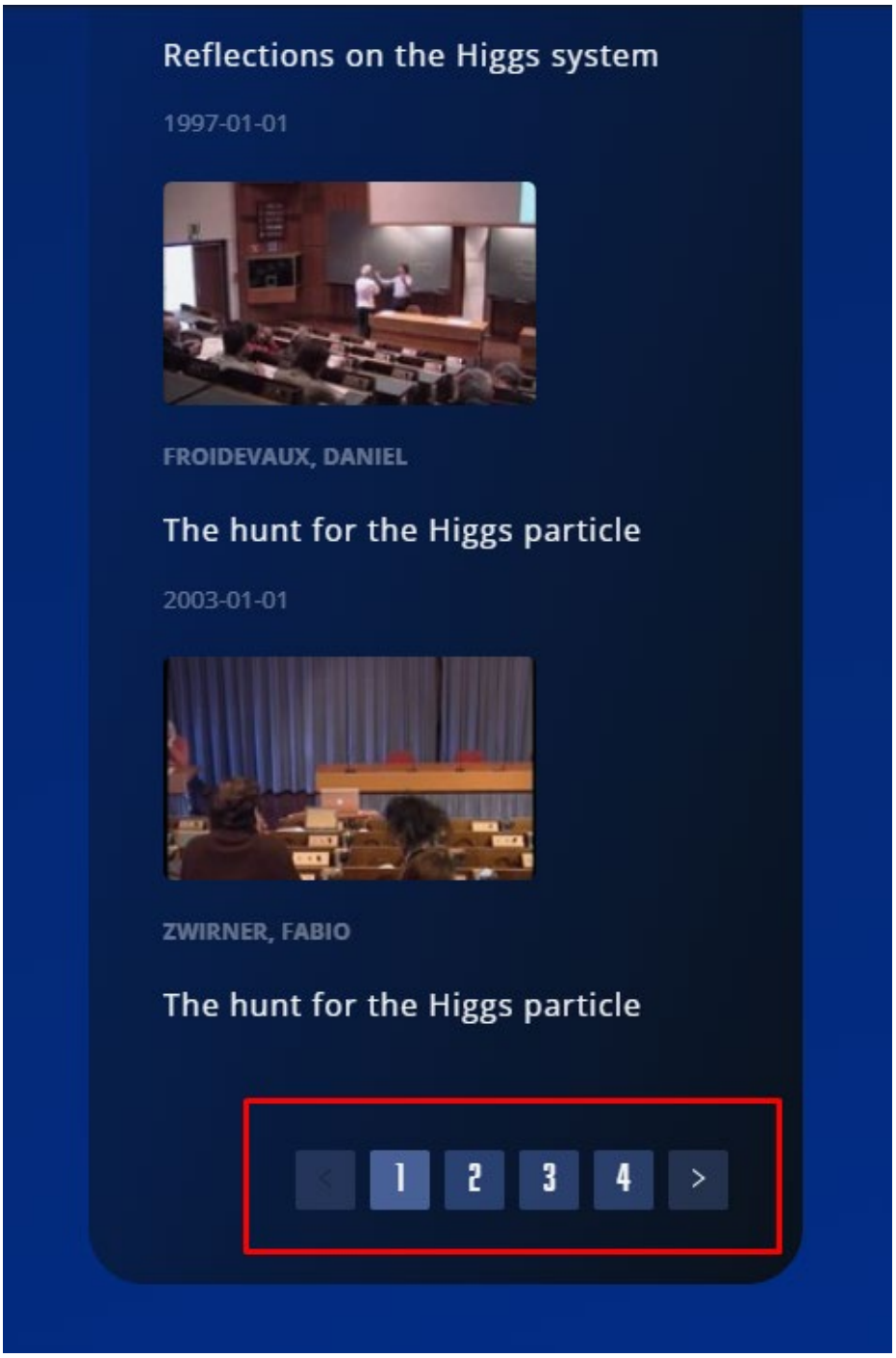
Note: You can find more help if you click on the (?) tooltip next to the search button.

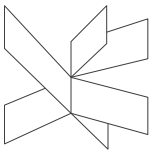


3. Find below the search results. Click on the lectures that you would like to check.



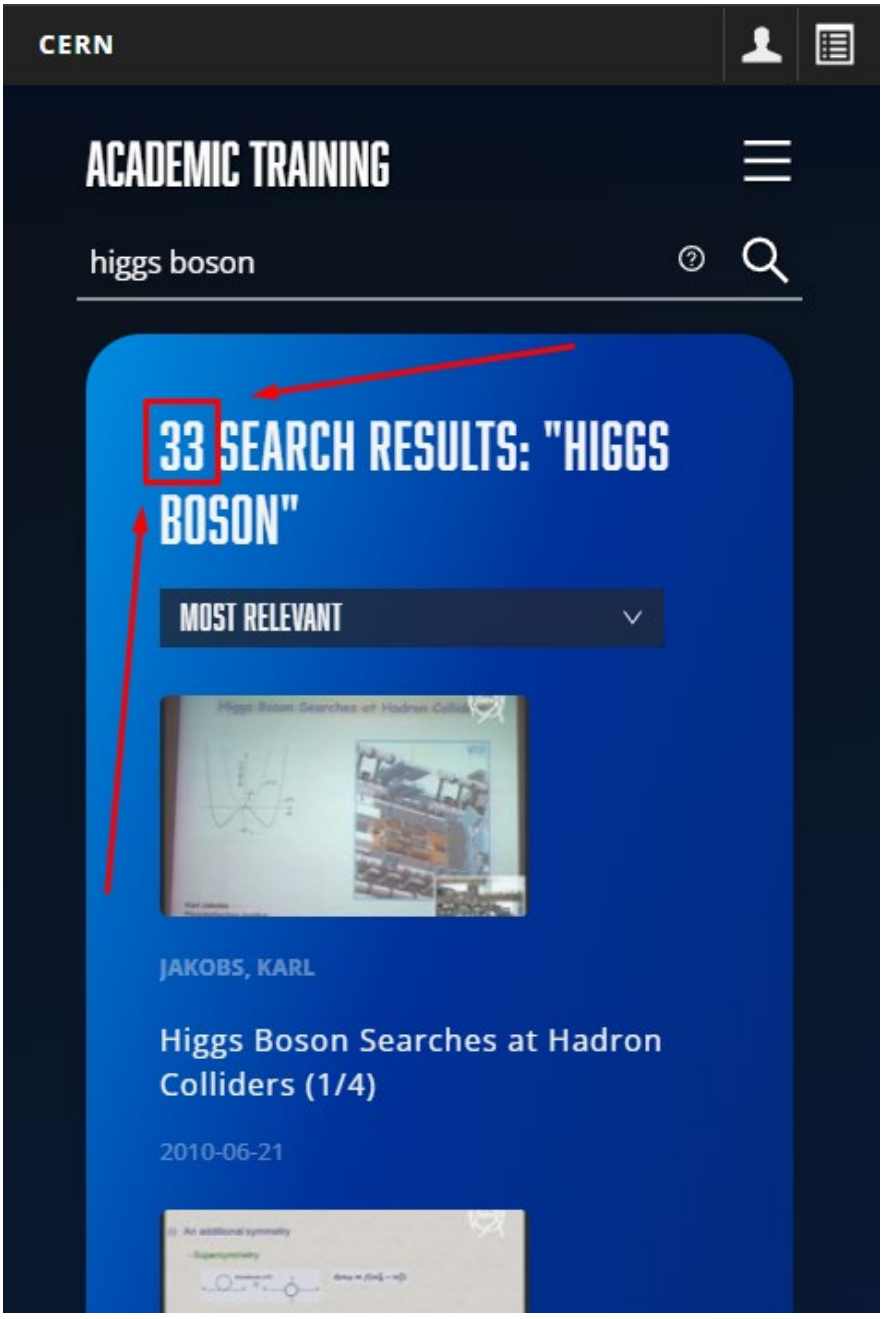
4. Scroll down for more results. Use the pagination to find even more lectures related to the searched term.

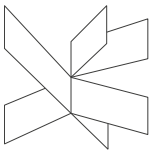




11 How to find the number of results from search?

The number of results can be found under the *Search Results* on the *Search Page*.

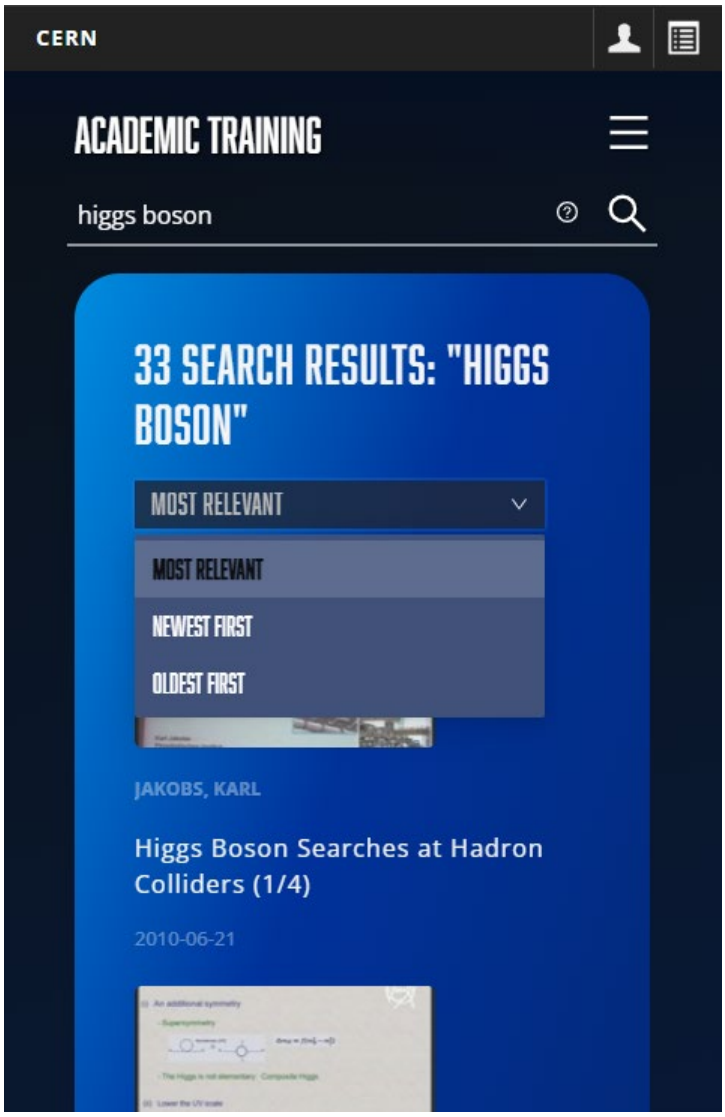


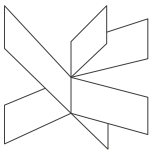


12 How to sort the search results by relevance?

If you would like to sort the lectures by relevance, you must click on the dropdown menu under the *Search Results* on the *Search Page*. The following is going to be listed:

1. Most relevant
2. Newest first
3. Oldest first

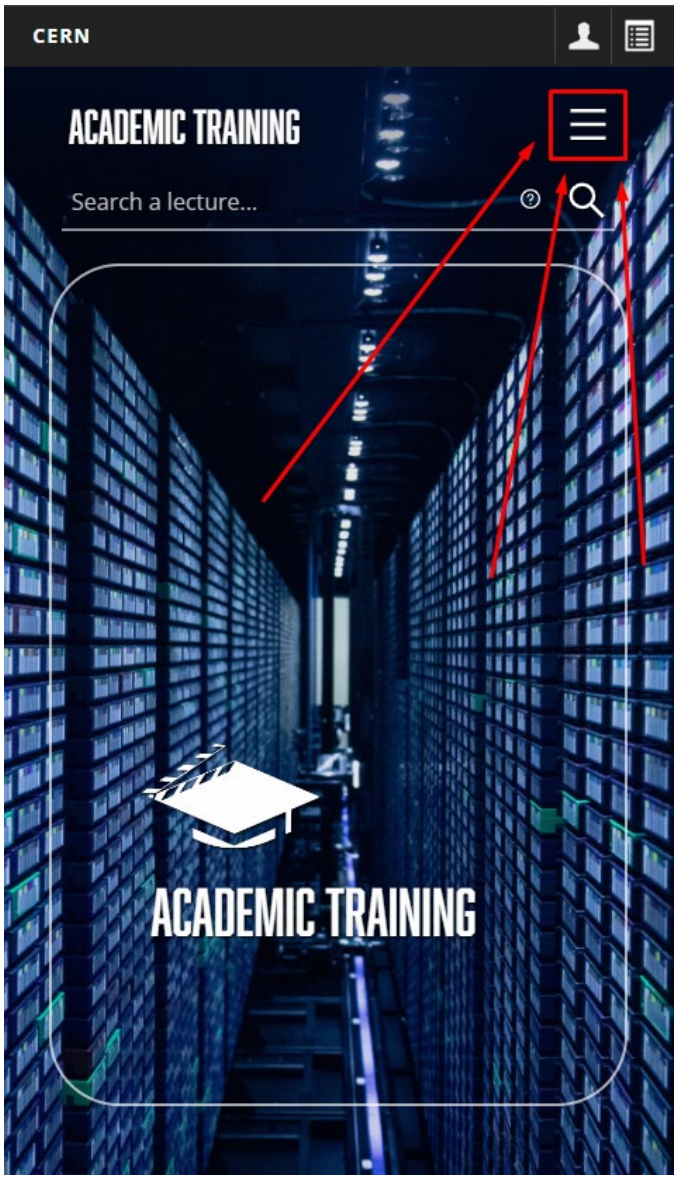




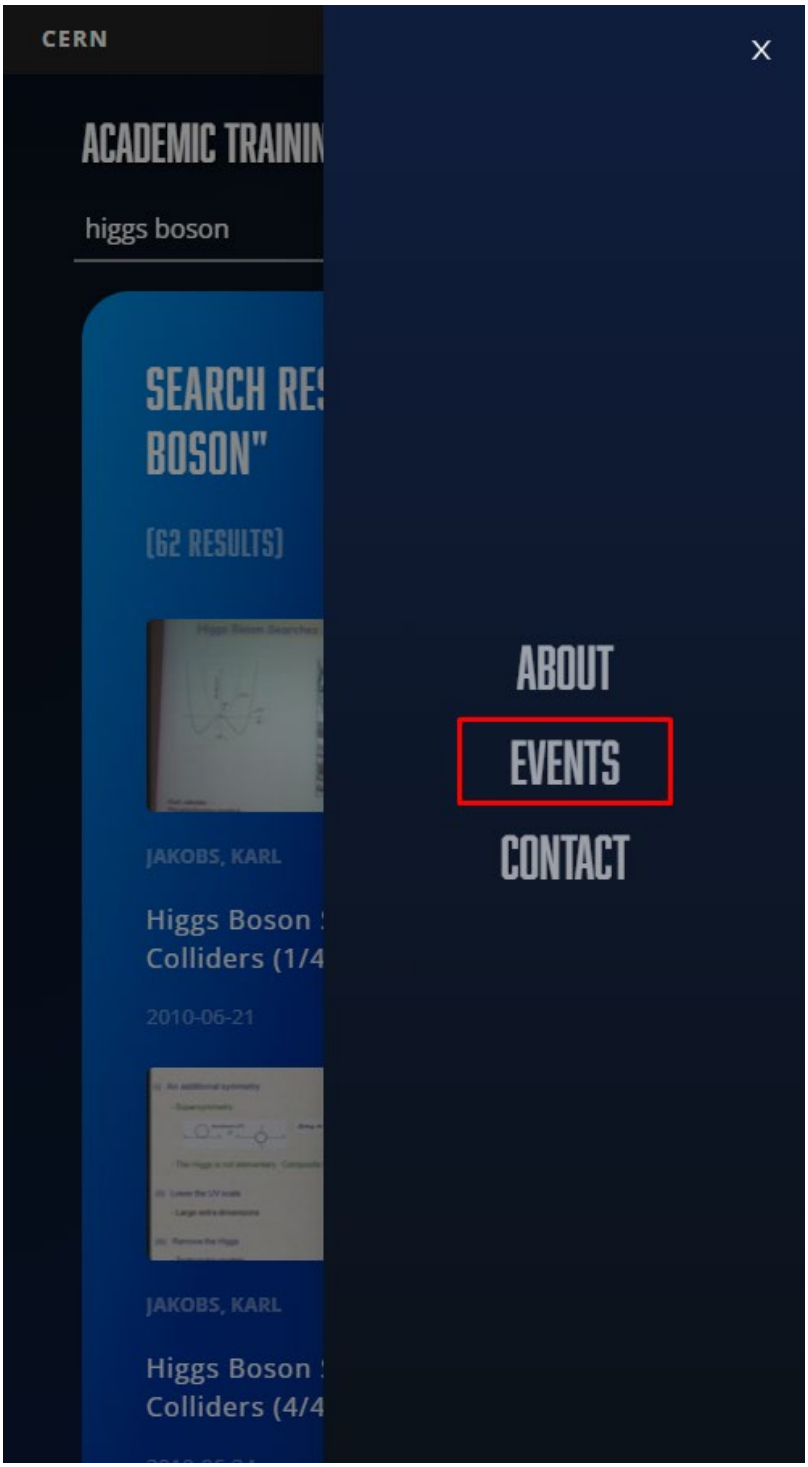
13 How to access the full agenda of the Academic Training lectures?

There are two ways to find the full programme of the Academic Training lectures.

1. Go to the header and find the ≡ button. Click on it, and a drawer menu should show up.

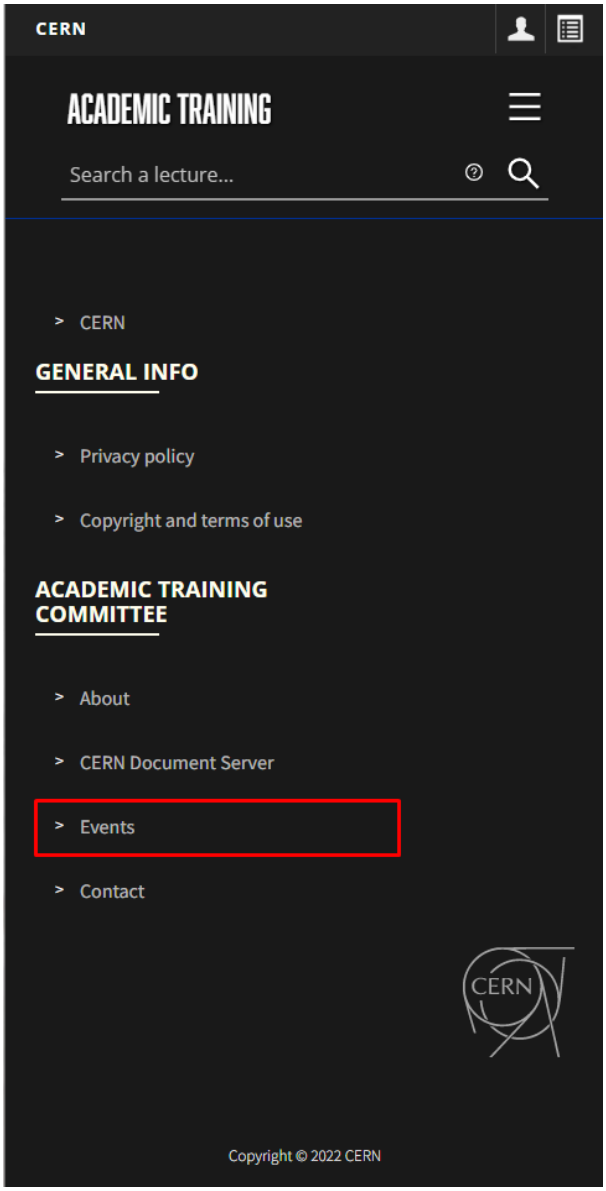


Once the Drawer is open, find the **EVENTS** button and click on it.



OR

2. Scroll down to the footer and find the **Events** button under the Academic Training Committee block. Click on it.

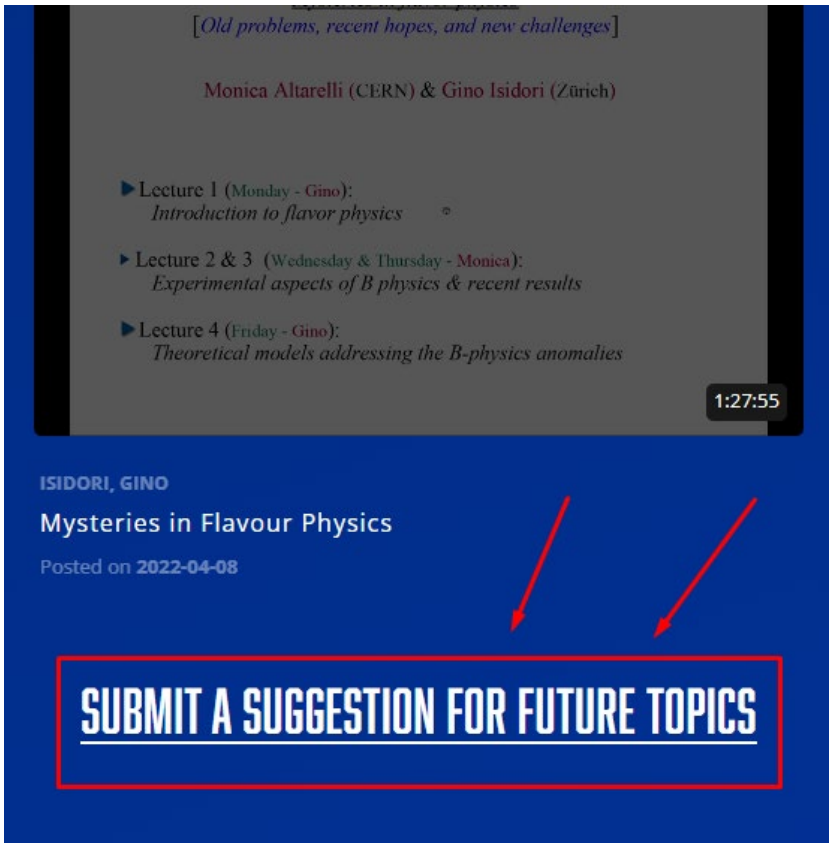


It should navigate you to the Indico page of the full *Academic Training Lecture Regular Programme*.

14 How to contact the ATC?

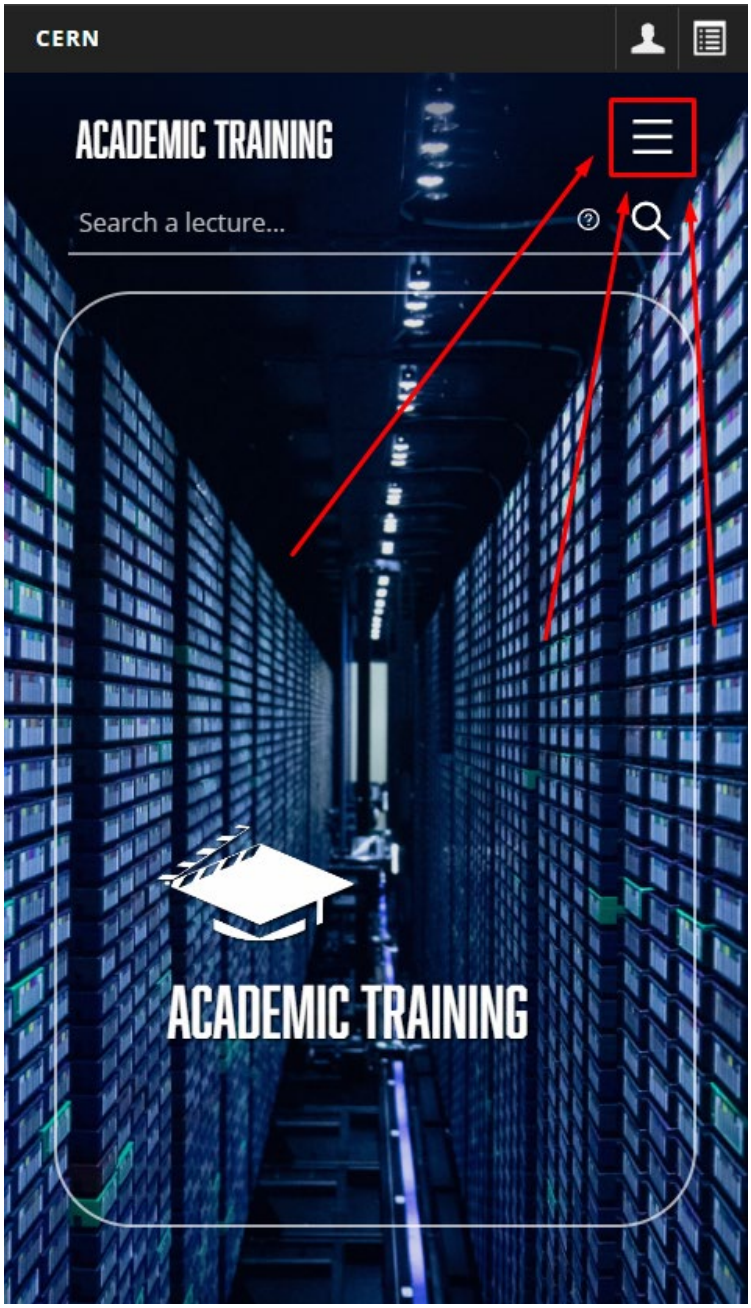
There are three ways of contacting the ATC (*Academic Training Committee)*.

1. Go to the *Main Page*, scroll down and find the **SUBMIT A SUGGESTION FOR FUTURE TOPICS** button. Click on it.

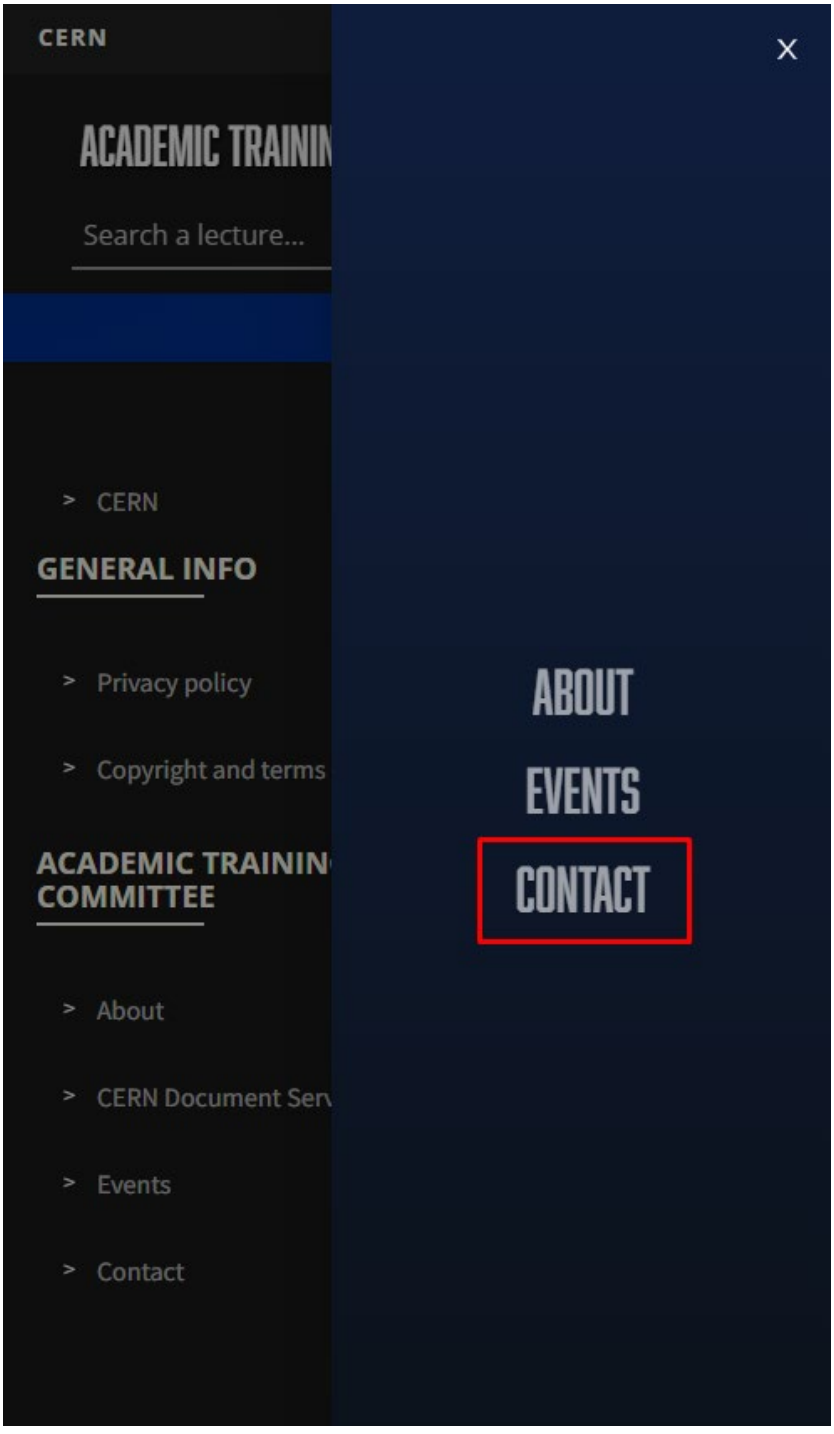


OR

2. Go to the header and find the ≡ button. Click on it, and a drawer menu should show up.

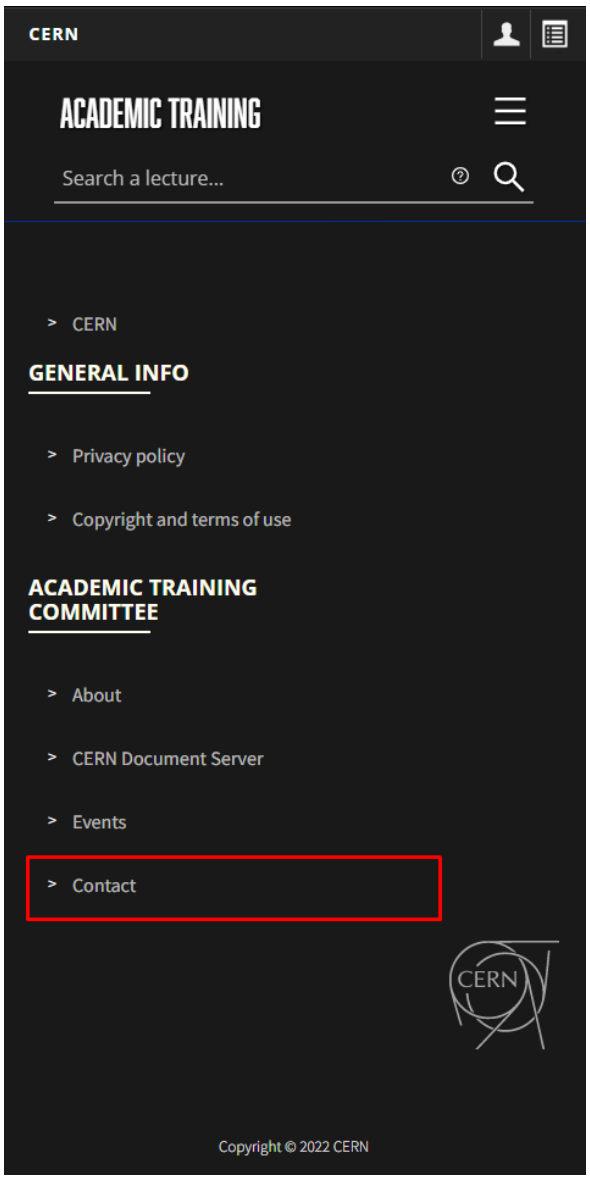


Once the Drawer is open, find the **CONTACT** button and click on it.



OR

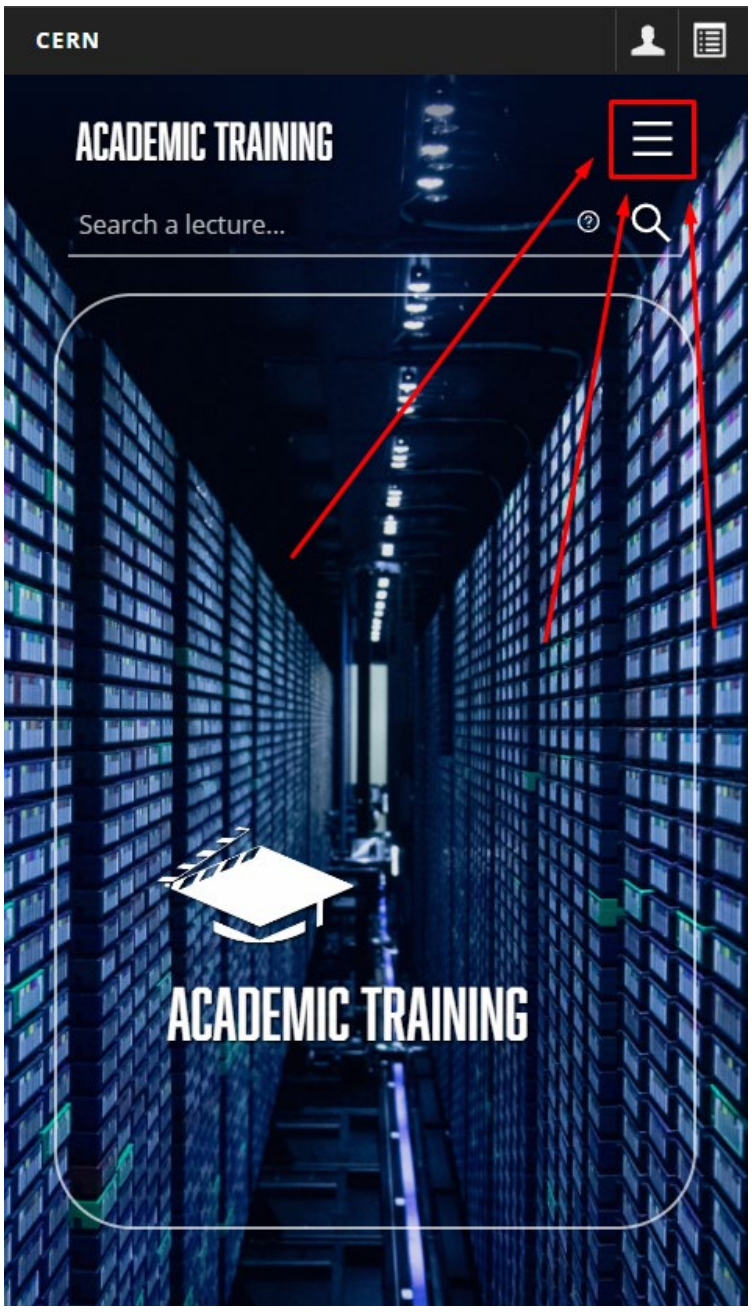
3. Scroll down to the footer and find the **Contact** button under the Academic Training Committee block. Click on it.

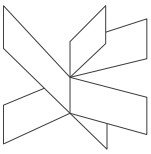


All of the above shall open your default email agent with the address *atc-contact@cern.ch*.

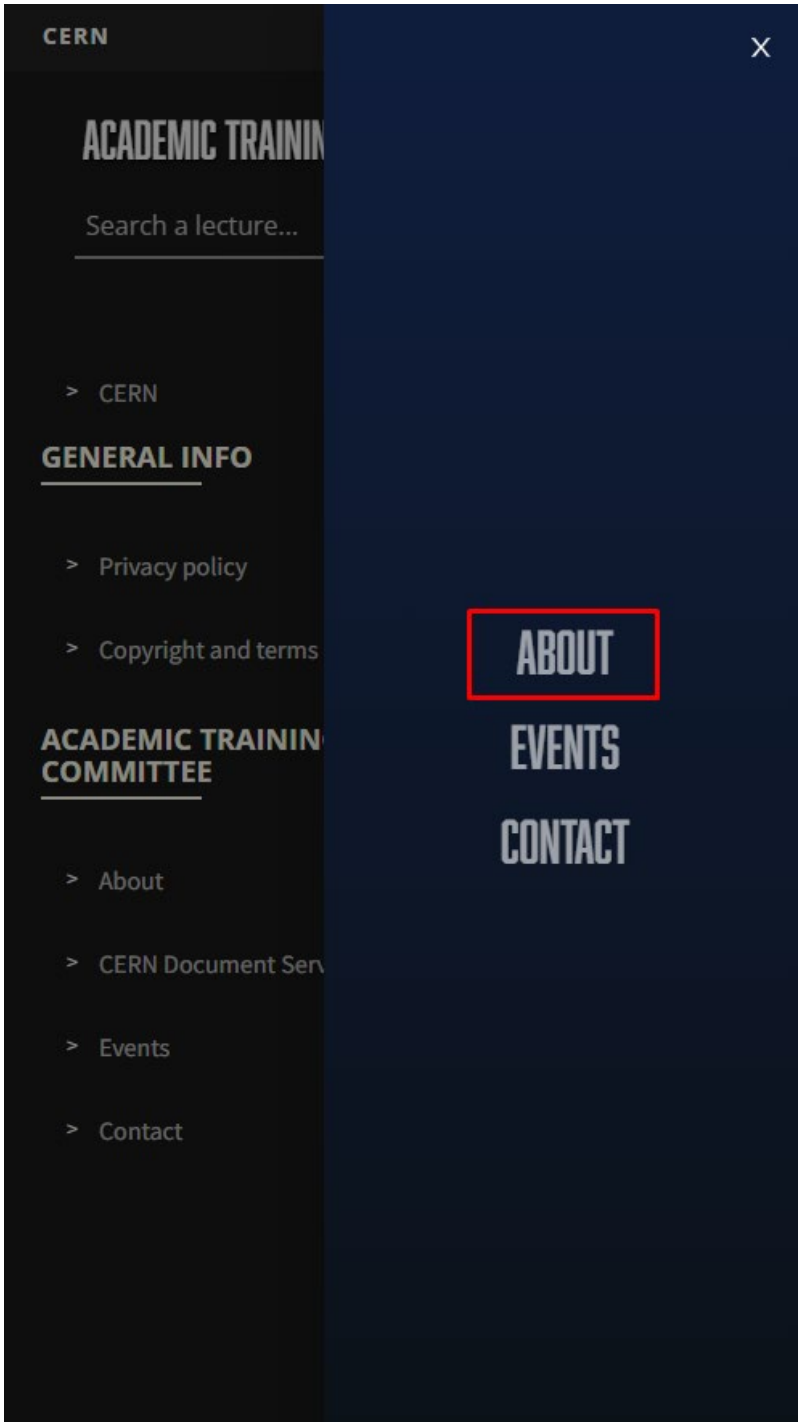
15 How to view who the members of the ATC are?

1. Go to the header and find the ≡ button. Click on it, and a drawer menu should show up.



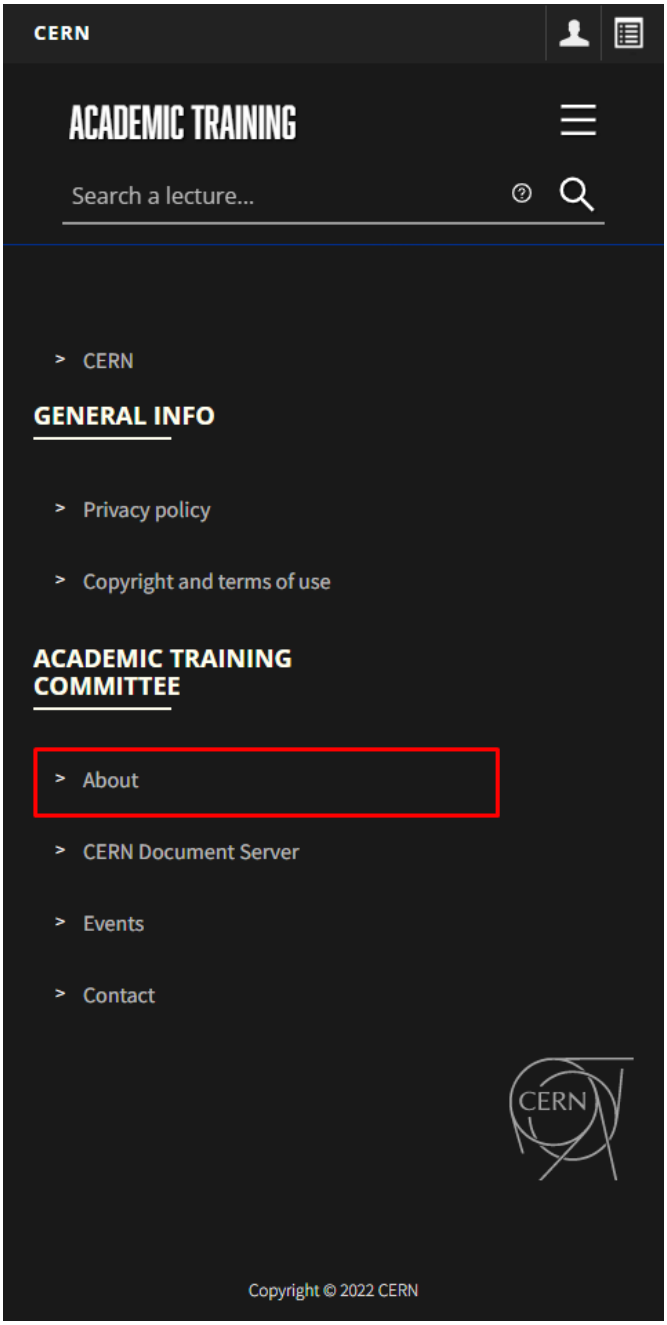


Once the Drawer is open, find the **ABOUT** button and click on it.



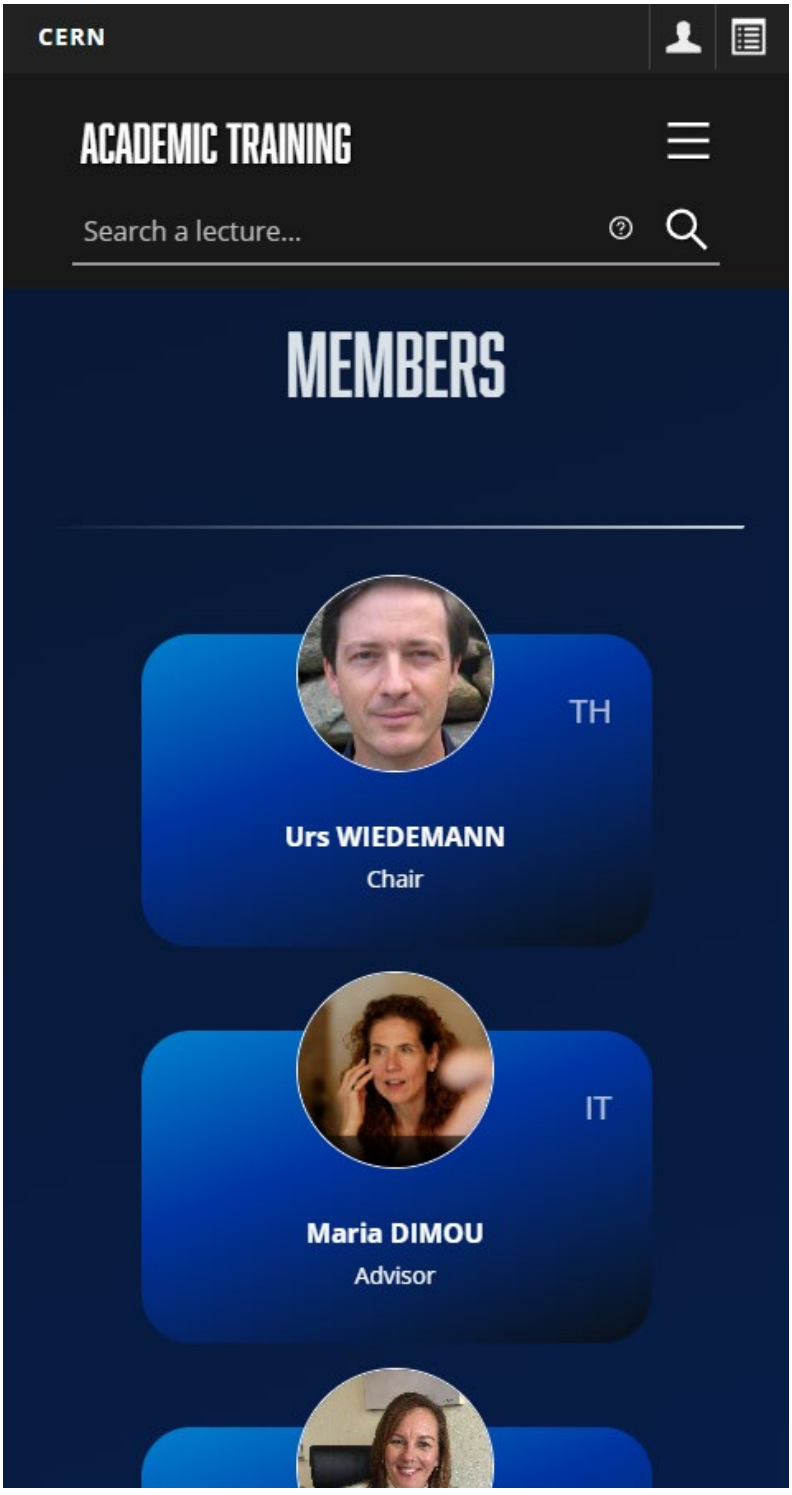
OR

2. Scroll down to the footer and find the **About** button under the Academic Training Committee block. Click on it.



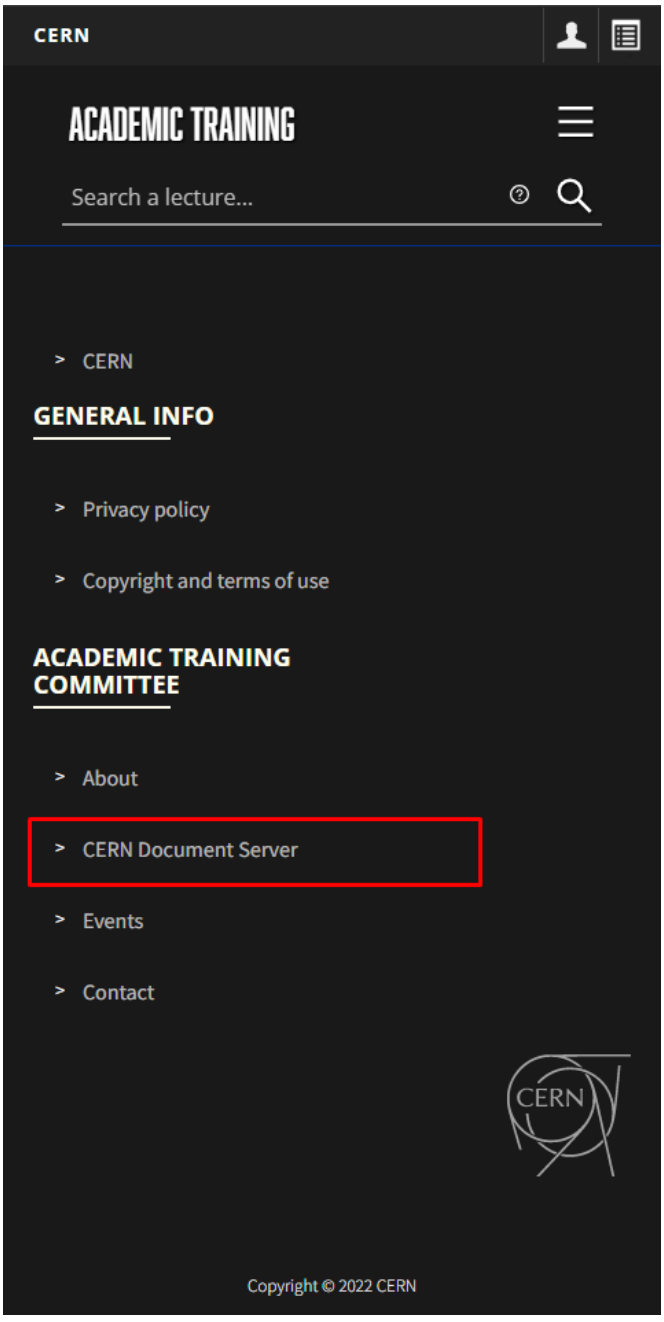
All of the above shall open the *About Us Page* that displays the current members of the Academic Training Committee.

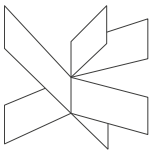
Scroll down to see the full list.



16 How to view the lectures from the CERN Document Server?

Scroll down to the footer and find the **CERN Document Server** button under the Academic Training Committee block. Click on it.





BSc Thesis in Software Technology Engineering

UI Design Choices

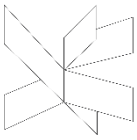
Building a website to promote the CERN Academic Training lectures

Franciska-Leonóra Török, 293171 IT

Supervisors:

Kasper Knop Rasmussen
VIA University College

Maria Dimou
CERN IT, Academic Training



VIA University College

2,701 characters

Software Technology Engineering

7th Semester

2023

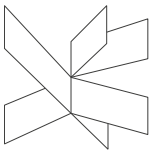
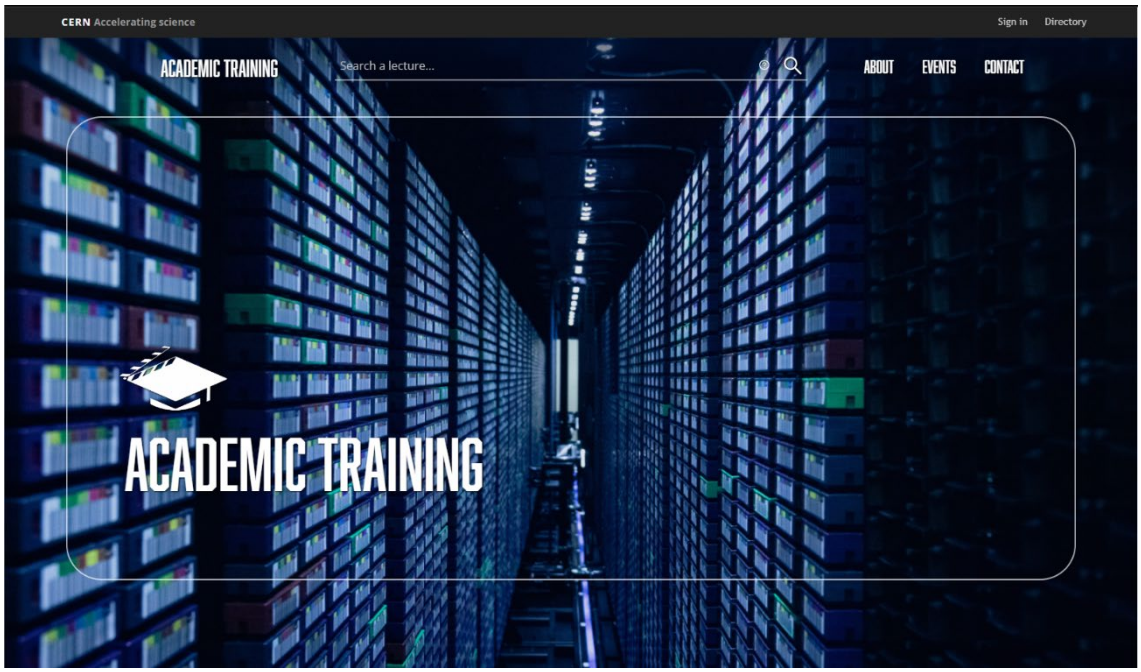


Table of content

1 Introduction

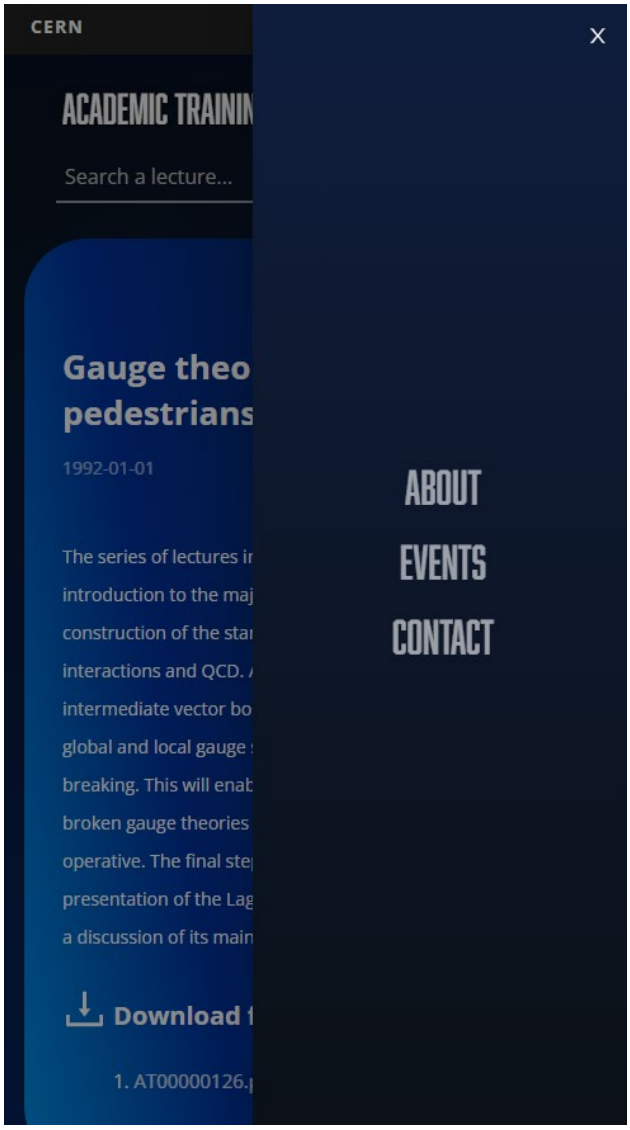
A carousel can be seen with several photos automatically switching vertically on the *Main page* of the website. The logo and the photos are specifically made for this project and have been taken around CERN. The photos can be accessed from **CDS**. The same carousel can be found on the *About Us page* as well.



The CERN toolbar and the Academic Training header with the search bar and the menu (About, Events, Contact) are permanently visible, even after scrolling.

On smaller screens (like mobile, tablet), the CERN toolbar and the Academic Training header take another view. The menu collapses into a hamburger icon and the search bar goes below it and the title. Upon clicking the hamburger icon, the menu will appear in a drawer.

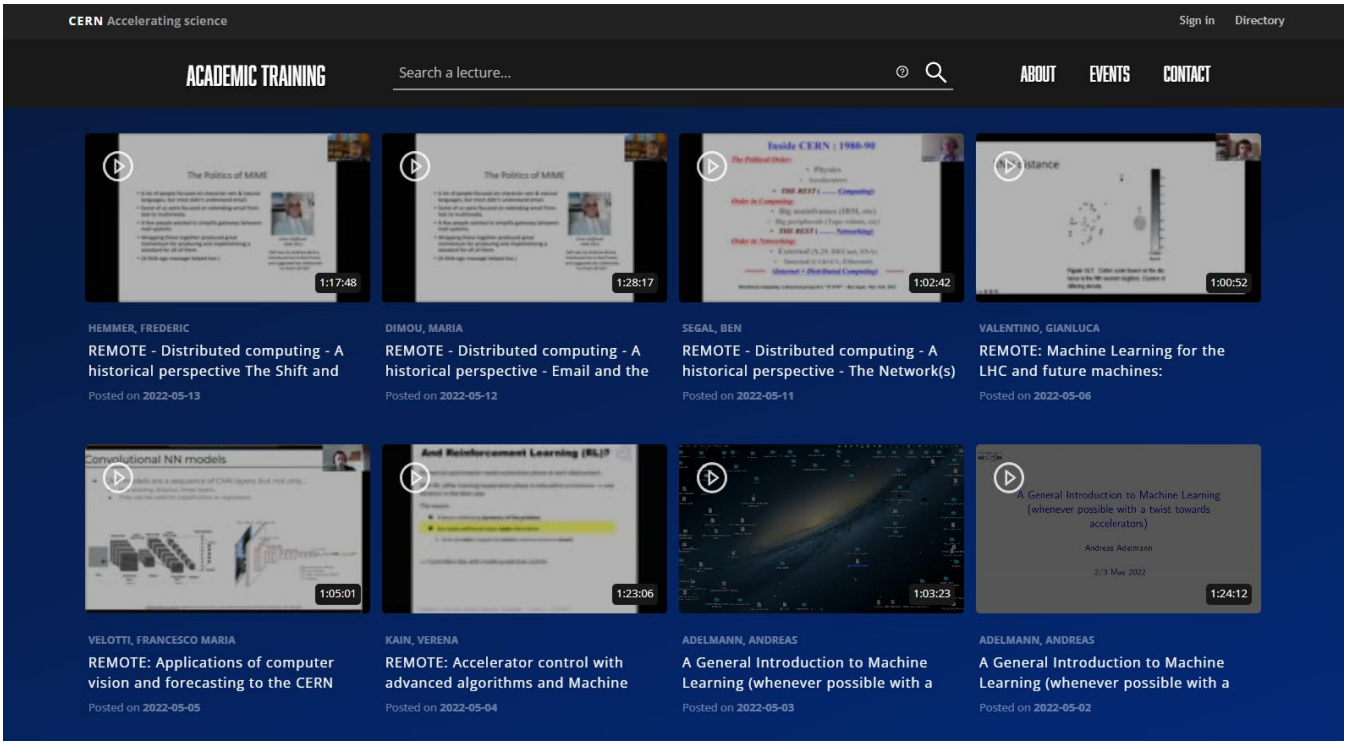

CERN
ACADEMIC TRAINING
Search a lecture...
ACADEMIC TRAINING


CERN
ACADEMIC TRAININ
Search a lecture...
Gauge theo
pedestrians
1992-01-01
The series of lectures i
introduction to the maj
construction of the sta
interactions and QCD.
intermediate vector bo
global and local gauge
breaking. This will enab
broken gauge theories
operative. The final ste
presentation of the Lag
a discussion of its main
Download
1. AT00000126.
ABOUT
EVENTS
CONTACT

2 The logo

The Academic Training logo represents an academic hat with the director's cut. The logo has been created with Adobe Illustrator ®.

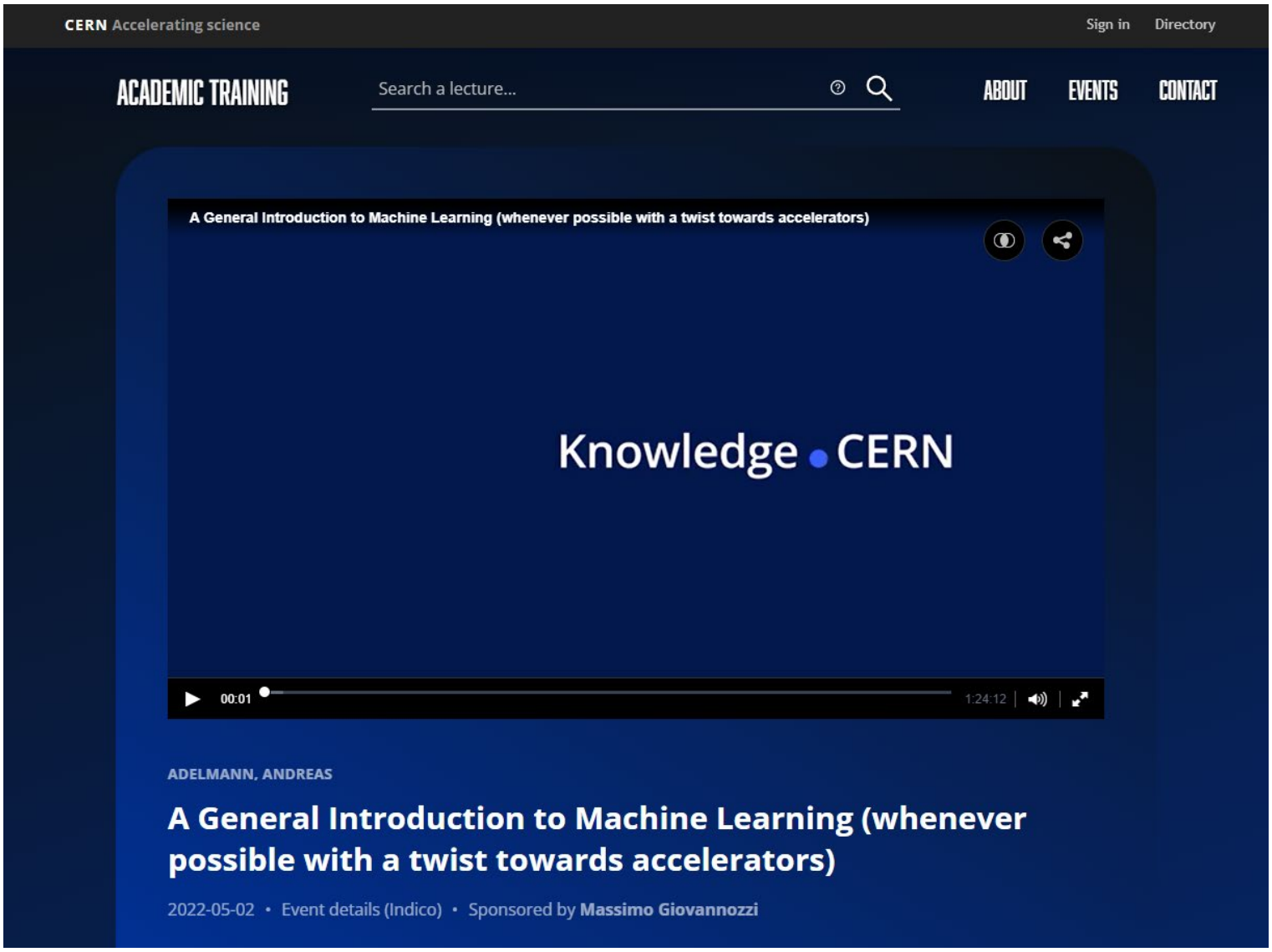
3 View lecture functionality



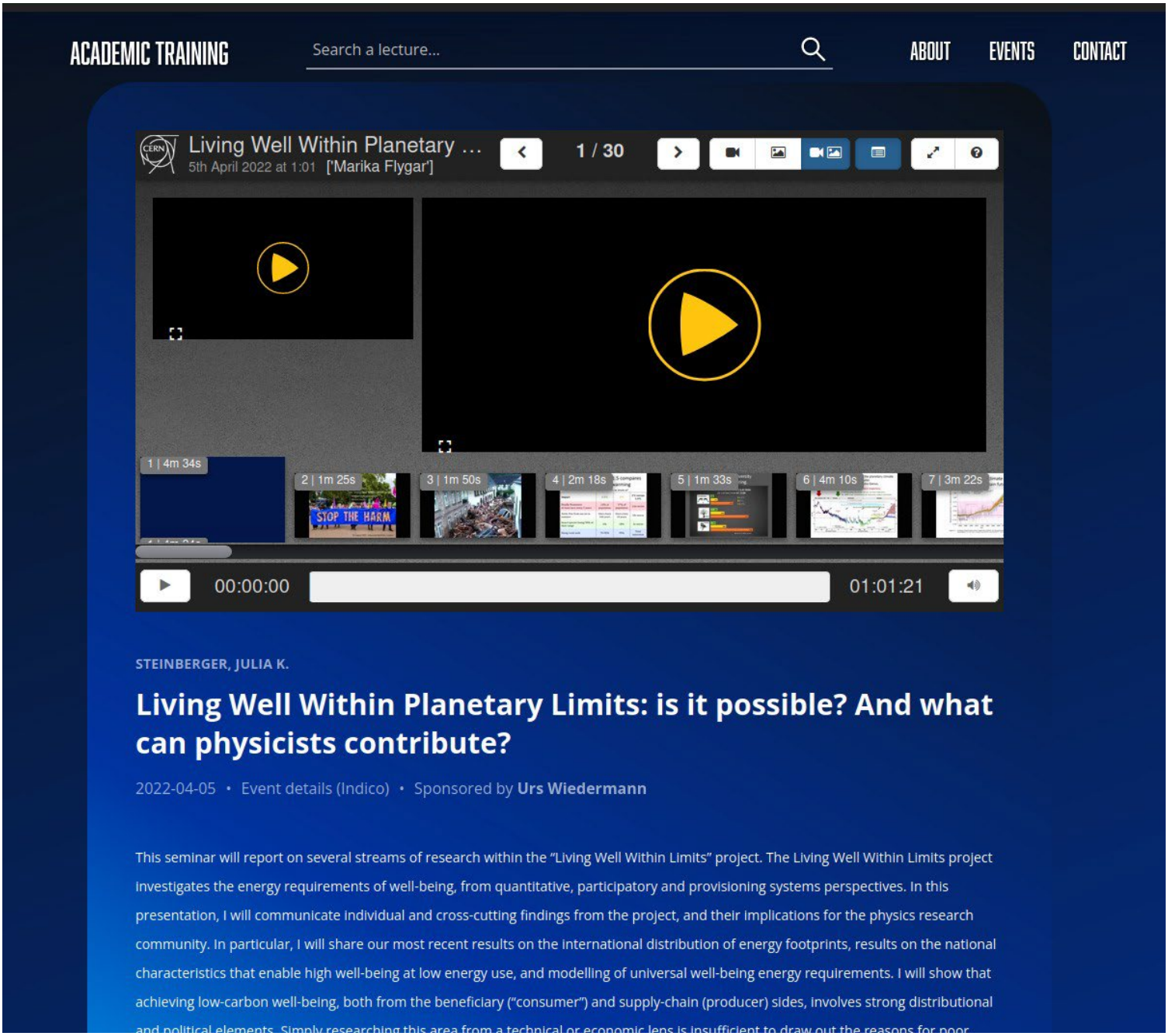
As it can be seen, a grid of videos will be displayed chronologically on the *Main page*. Each video lecture has the play button, thumbnail, speaker, title, release date, and the duration of the video.

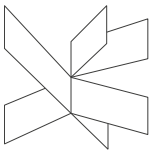
3.1 View video lecture

1. Single video



2. Two-channel video

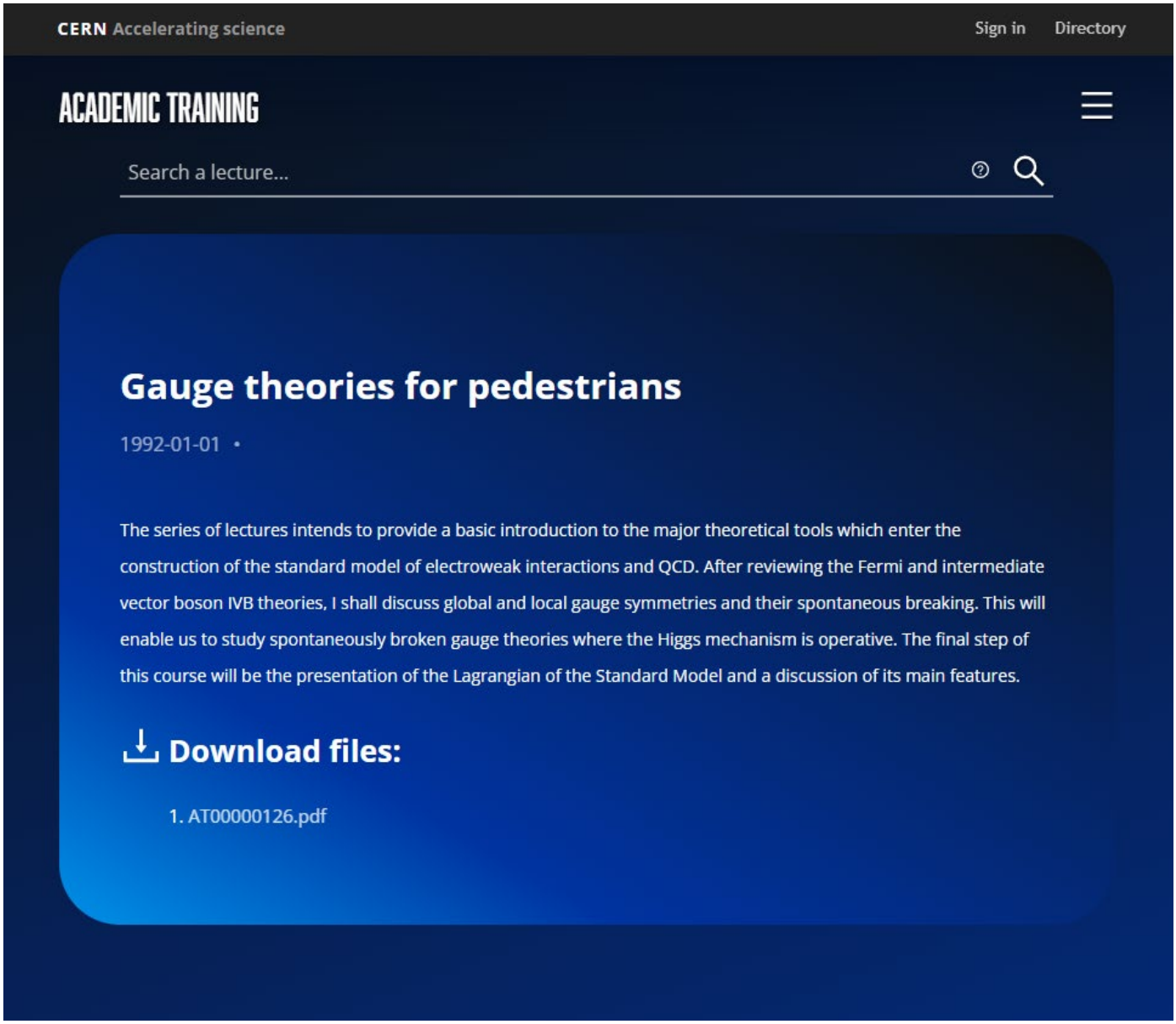




3.2 View non-video lecture

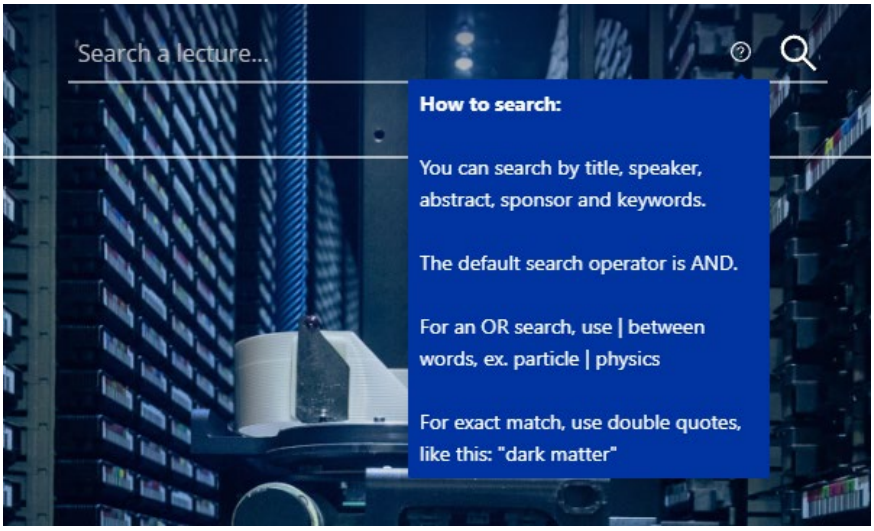
View list of files

Those lectures that have no video content but are available as downloadable files are displayed as follows:



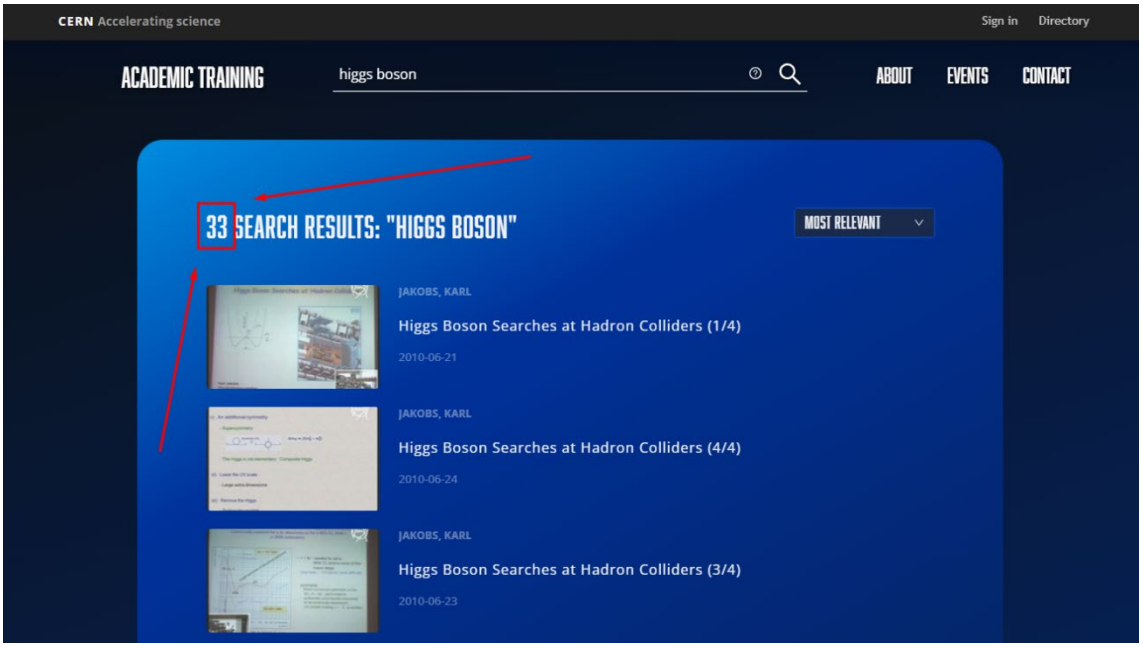
4 Search lecture functionality

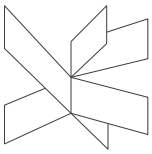
4.1 Help for search



4.2 View search results

1. View number of results

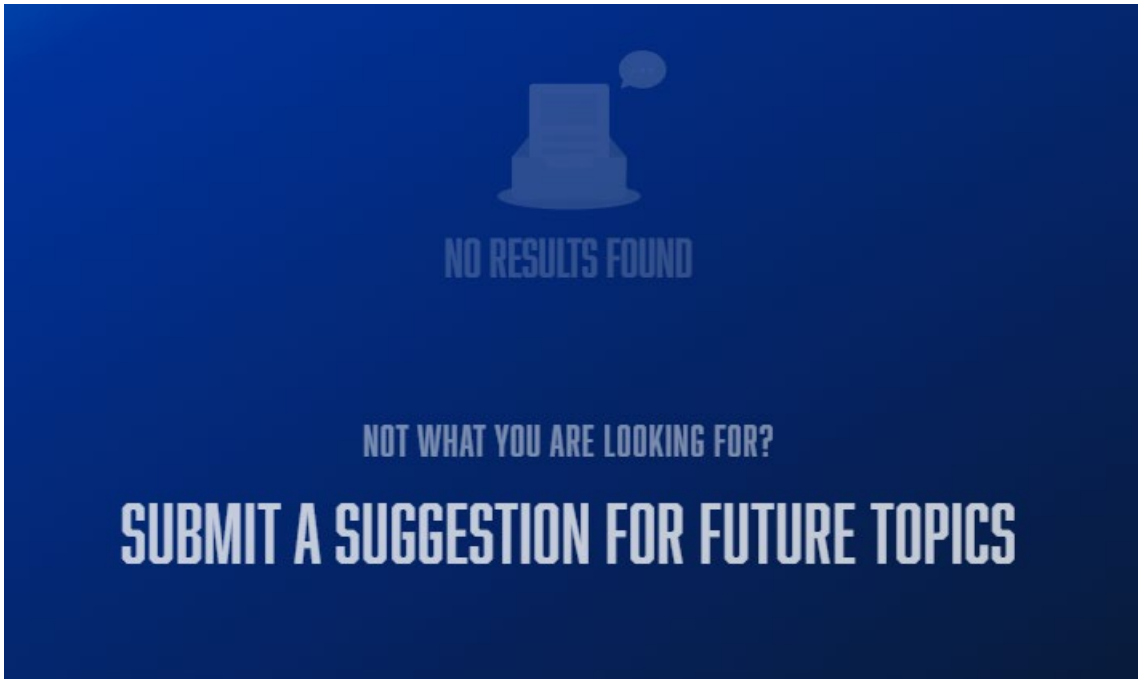




2. Sort results by relevance

4.3 Change search if no results are shown

If there are no results for a searched term, then the following appears:

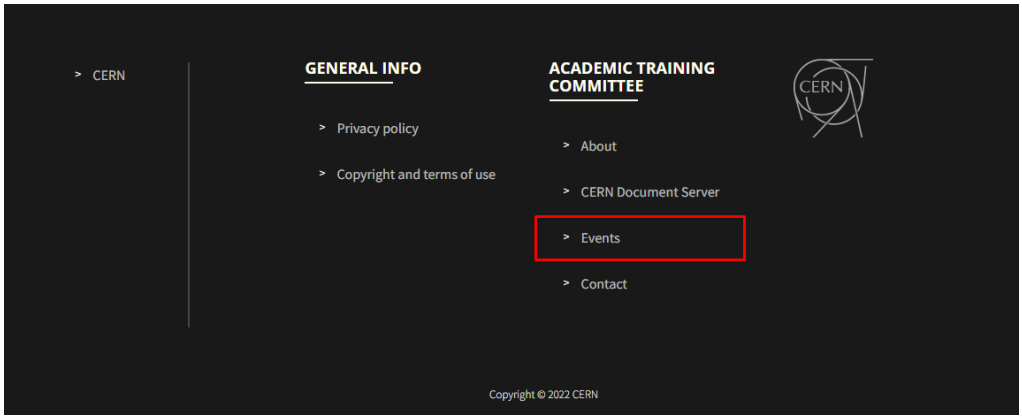


5 View ATC members functionality

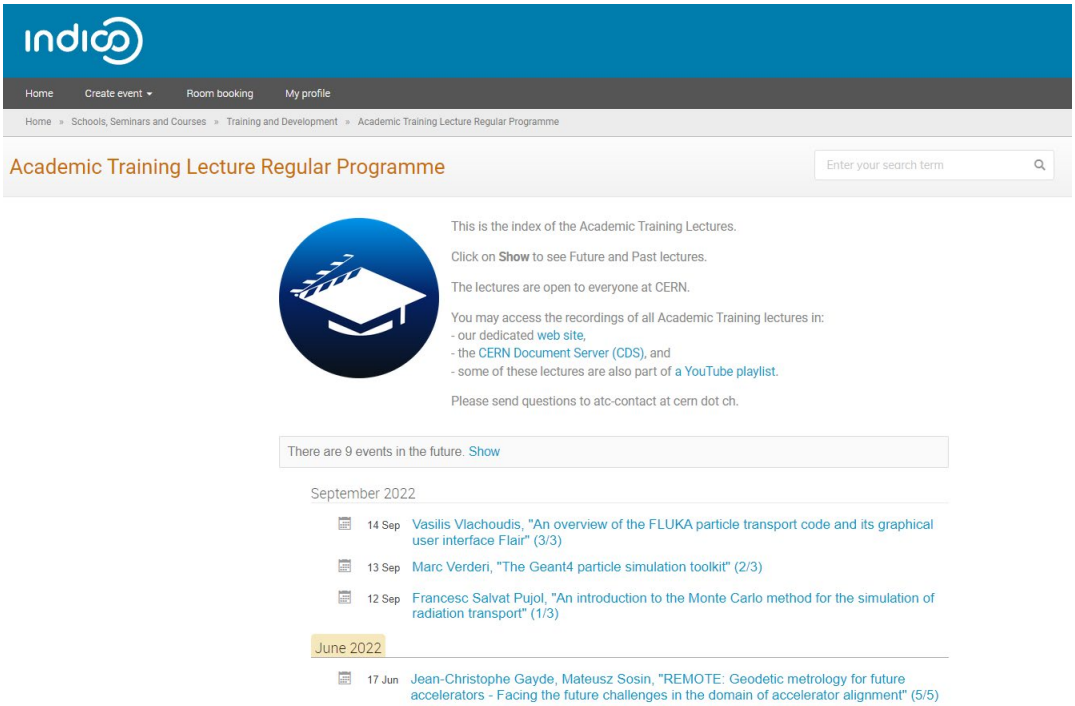
6 View Events functionality

On the header:

On the footer:



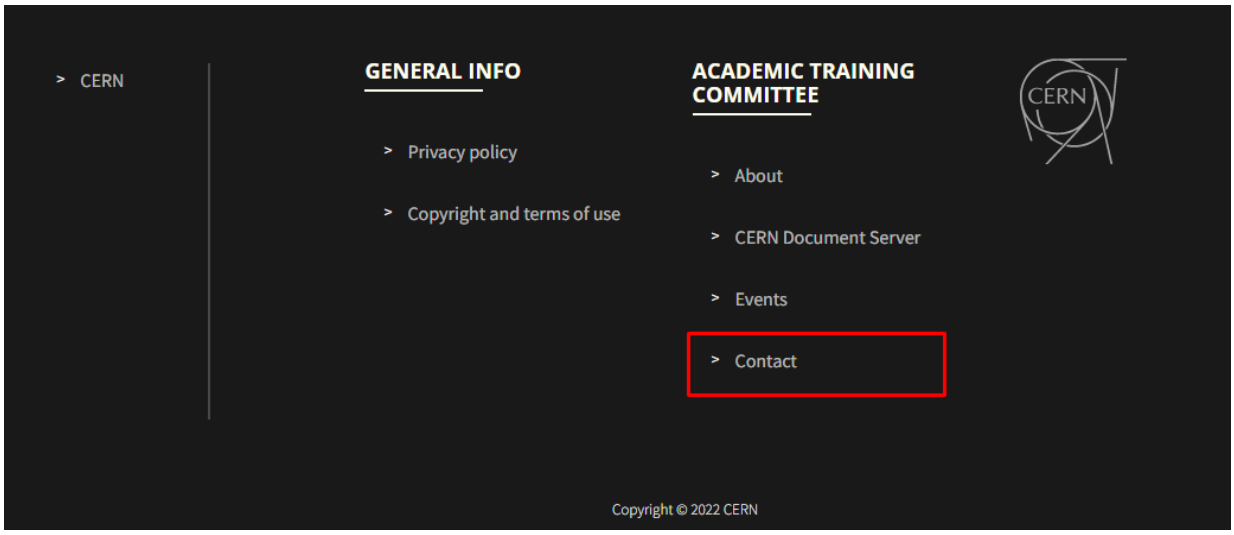
Both lead to the Full Programme of the Academic Training series that is available on Indico.



7 Contact ATC functionality

On the header:

On the footer:



A button that drives users to send suggestions for future lectures also leads to the contact of the ATC.