

## Copyright Warning

This paper is protected by copyright. You are advised to print or download **ONE COPY** of this paper for your own private reference, study, and research purposes. You are prohibited from having acts infringing upon copyright as stipulated in Laws and Regulations of Intellectual Property, including, but not limited to, appropriating, impersonating, publishing, modifying, altering, mutilating, distorting, reproducing, duplicating, displaying, communicating, disseminating, making derivative work, commercializing, and converting to other forms the paper and/or any part of the paper. The acts could be done in actual life and/or via communication networks and digital means without the permission of copyright holders.



Vietnamese-German University

The users shall acknowledge and strictly respect the copyright. The recitation must be reasonable and proper. If the users do not agree to all of these terms, do not use this paper. The users shall be responsible for legal issues if they make any copyright infringements. Failure to comply with this warning may expose you to:

- Disciplinary action by the Vietnamese-German University.
- Legal action for copyright infringement.
- Heavy legal penalties and consequences shall be applied by the competent authorities.

The Vietnamese-German University and the authors reserve all their intellectual property rights.



# **A Massive Open Online Courses platform based on OpenedX.**

**Kieu Hoang Khoi**

11264 - 1235531

 Supervisors:  
Vietnamese-German University

**Dr. Tran Hong Ngoc**

**Dr. Huynh Trung Hieu**

Computer Science - Bachelor Thesis

Oct 04, 2023

# Declaration

I hereby declare that the present bachelor thesis, which is submitted to the Vietnamese-German University and Frankfurt University of Applied Sciences, is a product of my own work unless otherwise referenced. I further declare that this thesis has not been previously or concurrently submitted, in part or whole, at any other universities or institutions.



Kieu Hoang Khoi

11264 - 1235531

CS2016

# Acknowledgement

This bachelor's thesis cannot be completed without the support of so many people. Therefore, I would like to express my gratitude to everyone who helped me during the process of doing this thesis.

I am especially grateful to Prof. Tran Hong Ngoc and Prof. Huynh Trung Hieu for supervising and guiding my bachelor's thesis.

Finally, I want to offer my sincere thanks to all my friends, my family, and all of my professors, staff, and faculty members who have supported me during my 4 years at VGU and given me the opportunity to finish my thesis completely.



Vietnamese-German University

# Table of Contents

List of Figures .....	6
Abstract .....	8
Chapter 1: Introduction .....	9
Chapter 2: Basic Knowledge .....	11
2.1 Fundamental concept.....	11
2.1.1 MOOCs .....	11
2.1.2 Learning Management System (LMS).....	12
2.1.3 Content Management System (CMS).....	13
2.1.4 Web Server.....	14
2.1.5 Ubuntu.....	15
2.1.6 Docker.....	16
2.1.7 Python .....	18
2.1.8 Django.....	18
2.1.9 ReactJS.....	19
2.2 EdX platform users.....	20
Chapter 3: Methodology .....	21
3.1 Server .....	21
3.1.1 Ubuntu.....	21
3.1.2 Python .....	22
3.1.3 Docker and Docker Compose Installation .....	23
3.1.4 Django.....	23
3.1.5 CMS .....	24
3.1.6 LMS .....	24
3.1.7 Create admin account.....	25



Vietnamese-German University

3.2	Instructor .....	25
3.2.1	XBlocks.....	25
3.2.2	LTI .....	26
3.2.3	API.....	26
3.2.4	Create course.....	27
3.2.5	Create library .....	28
3.2.6	Create Problem.....	29
3.2.7	Create Exam.....	31
3.3	Learner .....	33
3.3.1	Enroll a Course .....	33
3.3.2	Explore Dashboard.....	34
3.3.3	Check progress.....	36
3.3.4	Get Certificate.....	40
Chapter 4:	Results and Discussions .....	41
4.1	Server .....	41
4.1.1	Tutor.....	41
4.1.3	CodeJail.....	53
4.2	Instructor .....	55
4.3	Learner .....	55
Chapter 5:	Limitation and Future work .....	56
5.1	Limitation.....	56
5.2	Future work .....	56
Reference	.....	57



# List of Figures

Figure 1: MOOC .....	11
Figure 2: LMS features .....	12
Figure 3: LMS Applications .....	13
Figure 4: CMS feature .....	13
Figure 5: CMS component.....	14
Figure 6: Ubuntu Server features.....	16
Figure 7: Docker Structure.....	17
Figure 8: Django components.....	19
Figure 9: Yola program use edX.....	20
Figure 10: YOLA with edX.....	21
Figure 11: Create library.....	28
Figure 12: Authn page .....	42
Figure 13: Account page.....	42
Figure 14: Communication page.....	43
Figure 15: Course Authoring page.....	43
Figure 16: Discussion page.....	44
Figure 17: Gradebook page.....	45
Figure 18: Learning page .....	46
Figure 19: ORA page .....	46
Figure 20: Profile page.....	47
Figure 21: Notes plugin .....	48
Figure 22: Credentials plugin.....	48
Figure 23: Indigo theme.....	49
Figure 24: Cairn overview .....	51

Figure 25: Cairn dashboard.....	51
Figure 26: Cairn analyze .....	52
Figure 27: Codejail plugin with success .....	53
Figure 28: Codejail failure state.....	54
Figure 29: Instructor page.....	55
Figure 30: Learner page in a course.....	55



## Abstract

In this thesis project, we will install OpenedX platform. OpenedX is a MOOCs platform. To build the OpenedX, we will need to install the following web-based tools: Learning Management System (LMS), Content Management System (CMS), Web Server, Python, Django, Ubuntu, Docker, and ReactJS. We will develop a webserver to host the MOOCs platform. We will also develop the following features for instructors: authorize courses, edit grades, post the schedule for the course. We will create the testing environment for the learners.



Vietnamese-German University

# Chapter 1: Introduction

Online learning has grown essential in recent years. The COVID-19 epidemic, educating remote, rural, and regional students, and the advancement of the 4.0 revolution in technology to support the demand for education [1]. The COVID-19 epidemic forced a number of schools and institutions to temporarily close. In many schools, colleges, and institutions, face-to-face instruction is no longer offered. Since the social gap is so important at this point, this will have a detrimental effect on educational activities. Educational organizations are looking for alternative solutions to handle this challenging situation. In order to prevent a disruption in education, this shutdown encouraged the expansion of online learning opportunities. The optimal way to provide online course materials, engage students, and do assessments has been discussed by many faculty. For children and young people living in regional and rural areas, it can sometimes be difficult to find nearby schools that offer the breadth of subjects they want to explore. For those children, online education may be a practical option and a viable substitute for having to leave their homes to attend school. Through cutting-edge technology, students in underdeveloped nations may access world-class information.



Organizations that had previously been reluctant to change would now accept modern technology as a solution to their problems. It was difficult for the educational sectors to deal with the situation at this time, and professional education in particular was especially difficult. Microsoft Teams, Zoom, and Google Meet are some of the technologies that can help with this issue and provide online instruction throughout the quarantine. Online e-learning is defined as learning experiences using various electronic devices with internet access (such as PCs, laptops, smartphones, etc.) in synchronous or asynchronous learning environments. Online education could serve as a platform to improve the flexibility, creativity, and student-centeredness of the educational process. When delivering content to students in rural and remote places, online course delivery is affordable and convenient. Organizations that had previously been reluctant to change would now accept modern technology as a solution to their problems [2]. It was difficult for the educational sectors to deal with the situation at this time, and professional education in particular was especially difficult. Microsoft Teams, Zoom, and Google Meet are some of the technologies that can help with this issue and provide online instruction throughout the quarantine. Online e-learning is defined as learning experiences using various electronic devices with internet

access (such as PCs, laptops, smartphones, etc.) in synchronous or asynchronous learning environments. Online education could serve as a platform to improve the flexibility, creativity, and student-centeredness of the educational process. When delivering content to students in rural and remote places, online course delivery is affordable and convenient. In the past, teachers would usually present classes using Prezi or other similar software, such as PowerPoint, in order to make them as short as possible [3]. However, advancements in education have coincided with technological advancement. Teachers can deliver live lectures to students at their homes while also recording the lectures for later study if necessary. Students can submit questions beneath the videos, and teachers can explain them at any time.

Universities and educational institutions are starting to use Massive Open Online Course education platforms for teachers to upload courses and manage students. OpenedX is one of several platforms that is an open-source course management system (CMS) that was originally developed by edX. The platform was originally developed as a research project at the Massachusetts Institute of Technology (MIT), with maintenance transferred to edX in 2012. On June 1, 2013, edX open-sourced the platform, naming it Open edX to distinguish it from the organization itself [4]. The source code can be found on GitHub. When edX was acquired in 2021 by 2U – an American educational technology company, the Open edX team, and maintenance were transferred to the Center for Reimagining Learning (tCRIL), a nonprofit founded by Harvard and MIT with the proceeds from the acquisition. In 2023, the nonprofit was renamed the Axim Collaborative [5]. The platform has been released one to two times a year since 2013. Each release is named after a tree, honoring the tree of knowledge (Lilac, Maple, Nutmeg, Olive, Palm). There are 250 leading institutions that have used this platform to upload their courses for online learning like Cambridge, Texas, ETH Zurich, ... like partners. In Vietnam, there are several educational centers that have begun to apply the platform to their system such as YOLO, and Lac Viet.

In this thesis report, chapter 1 is the Introduction. Chapter 2 is the Basic Knowledge. In Chapter 2, we will present the fundamental concept of Massive Open Online Courses (MOOCs). Chapter 3 is the Methodology. In Chapter 3, we will present the concept of the OpenedX platform with the Tutor environment. Chapter 4 is the Result and Discussions. In Chapter 4, we will present the output of the platform and the constraints during the project. In chapter 5, we will present the future works.

## Chapter 2: Basic Knowledge

In this chapter, we will present basic knowledge of MOOCs. Basic knowledge includes fundamental concepts and edX platform users. In the fundamental concept, we will present the idea of MOOCs in detail. In the edX platform users, we will present the current institutions that are using and applying the edX platform in their business.

### 2.1 Fundamental concept

In the fundamental concept, we will introduce the concept of MOOCs and the 8 web-based tools that make up MOOCs. The 8 web-based tools include Learning Management System (LMS), Content Management System (CMS), Web Server, Python, Django, Ubuntu, Docker, and ReactJS. We will present the above-mentioned parts of the MOOCs in detail.

#### 2.1.1 MOOCs

MOOCs are free online courses available for anyone to enroll in. These provide an affordable and flexible way to learn new skills, advance your career, and deliver quality educational experiences at scale.

In Figure 1, MOOCs are shown to have the following components: open registration, self-paced, start/end dates, credits, the role of the instructor, learning community, real-time

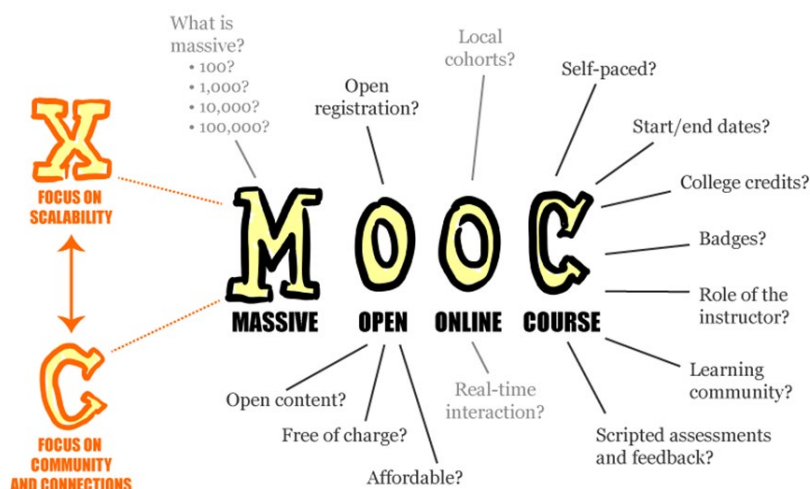


Figure 1: MOOC

interaction, free of charge/with the fee, and open content.

One of the open-source MOOCs available is called the OpenedX platform. The platform was originally developed as a research project at the Massachusetts Institute of Technology (MIT), with maintenance transferred to edX organizations in 2012 [5].

In this thesis, we will develop a MOOC platform based on OpenedX. To accomplish the components of OpenedX, we need to understand the web-based tools required to build the OpenedX platform. There are 8 web-based tools required. We will describe each tool in detail.

## 2.1.2 Learning Management System (LMS)

The Learning Management System (LMS) is a web-based tool used to create the OpenedX platform. LMS is the most visible part of the OpenedX. The LMS provides the instructor dashboard that users who have the Admin or Staff role. The LMS features are divided in 2 sections: one is for the learners and the other is for the instructors. Figure 2 show the features of the two sections. Learner features include access to the learner profiles, check the learner enrollment status, and browse courses via learner dashboard. Instructor features include publishing the instructor courses, post discussion boards, managing learner teams and cohorts, edit learner grades, and communicate with learners through discussion boards.

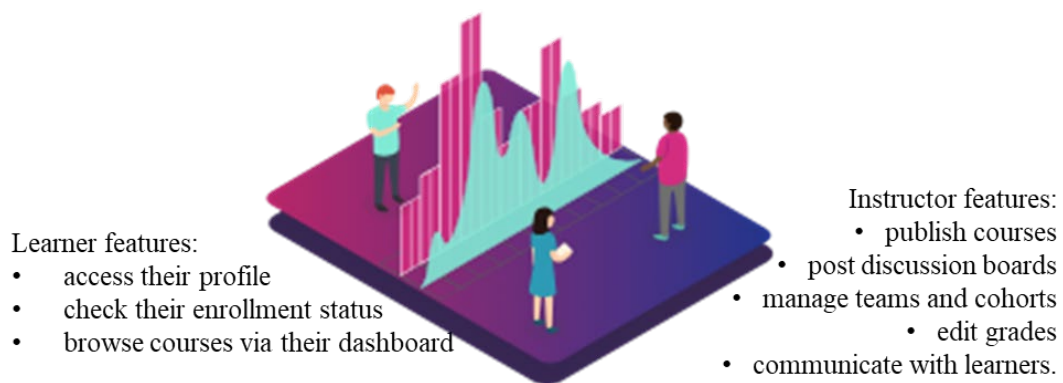


Figure 2: LMS features

There are many applications that utilize the LMS to develop their system. Figure 3 shows an example of various applications using LMS in their system.



Figure 3: LMS Applications

### 2.1.3 Content Management System (CMS)

The Content Management System (CMS), also called Studio, is a web-based tool used to provide the instructor feature for authorized courses. The CMS features are for instructors only. Figure 4 shows the features for instructors include author and edit courses content, manage instructors course schedule, course learner team, import videos, pictures with rich 3<sup>rd</sup>-party tools and add additional building blocks.



- Author courses and evolve your content while live with learners
- Manage your course schedule, course team, and grading policy
- Utilize documented and open XML standards (OLX) for import/export of courses.
- Easily import content and manage content across platforms.
- Access rich 3rd-party tools or add additional building blocks

Figure 4: CMS feature

There are many components that are included in CMS. Figure 5 shows CMS components used in system.

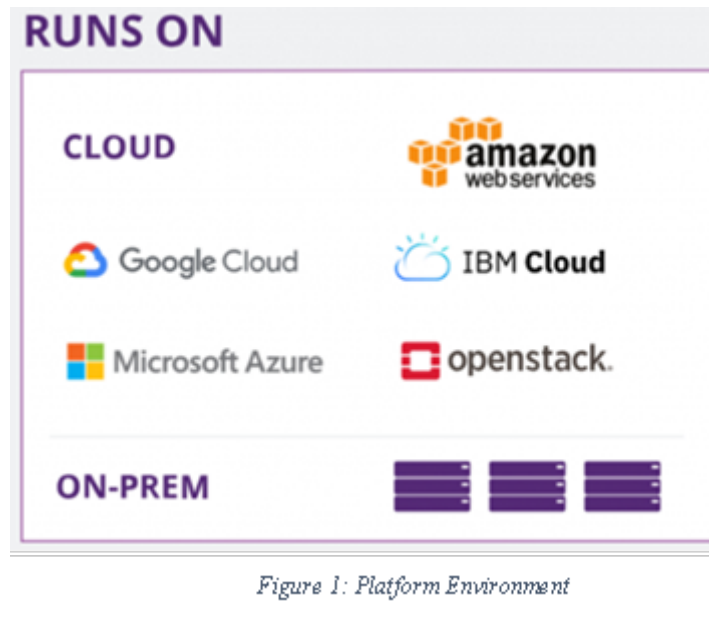


Figure 5: CMS component

## 2.1.4 Web Server

Webserver is a software and hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. The main job of a web server is to display website content through storing, processing and delivering webpages to users. Besides HTTP, web servers also support SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol), used for email, file transfer and storage.

The Open edX learner platform can operate on many platforms such as on web services cloud (Google, Amazon, ...) or on-premise (Server systems). With our project, we use the on-prem option to manage the data offline to host, store, manage, and distribute websites, web applications, and other digital content. Web servers use the Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) to communicate with web browsers. When a user types in the URL of our website, the browser locates our server, reads the client request, and processes the data. It is a necessary backbone for countless everyday tasks across the web. They are used to host e-commerce websites, deploy web applications for data management, and distribute multimedia content like videos and images. Figure 6 shows the two types of web server, one is on cloud and the other is on prem.



### 2.1.5 Ubuntu

Ubuntu is a modern open-source operating system based on Linux designed for enterprise servers, desktops, cloud, and IoT. **Ubuntu Server** is a version of the Ubuntu operating system designed and engineered as a backbone for the internet.

Ubuntu Server brings economic and technical scalability to your data center, public or private, that makes data or services available to other computers or programs on a network. It refers to an optimized version of the Ubuntu operating system that is specifically developed for server use in the context of the Ubuntu Server. It is designed for server activities including web hosting, file hosting, and application hosting. It includes a number of server-oriented programs, such as Apache HTTP Server, Nginx, and PHP, which are commonly used for web serving and web application development.

During the installation of the Ubuntu Server, I had to set up the SSH. The default Ubuntu install has no open ports. It is very common to administer servers via SSH, so the installer allows it to be installed with the click of a button in the OpenSSH setup. I can also import keys for the default user from GitHub or Launchpad to store the code in my repositories. I also installed some featured server snaps during the installation like docker, PowerShell, micro8s, ... Figure 7 show many feature that needed in running an Ubuntu Server such as Docker which will provide in detail later.

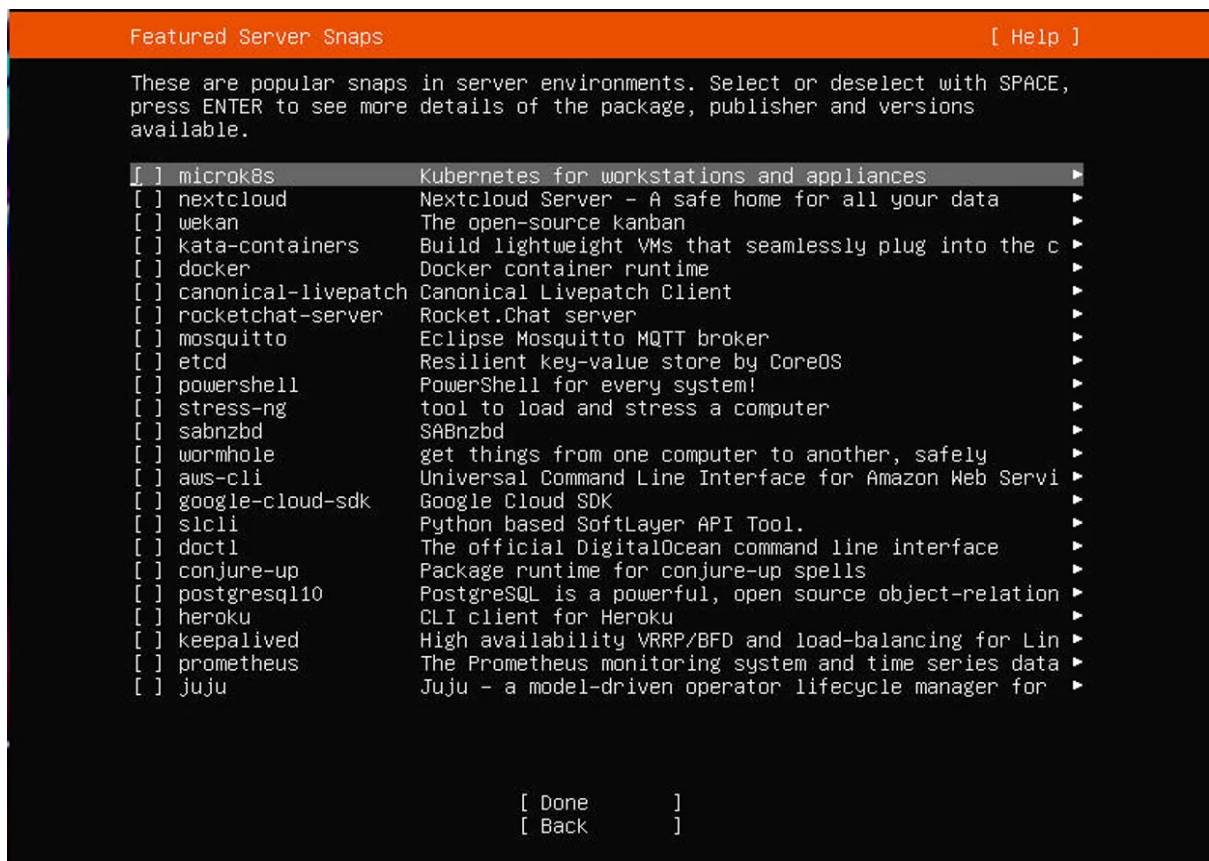



  
 Figure 6: Ubuntu Server features  
 Vietnamese-German University

## 2.1.6 Docker

### Docker

Docker is a platform designed to help developers build, share, and run container applications. Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing our Docker containers. The Docker client and daemon can run on the same system, or we can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface. Another Docker client is Docker Compose, which lets we work with applications consisting of a set of containers. The software that hosts the containers is called **Docker Engine**.

### Docker Engine

Docker Engine is an open-source containerization technology for building and containerizing our applications. Docker Engine acts as a client-server application with:

- A server with a long-running daemon process [dockerd](#).

- APIs which specify interfaces that programs can use to talk to and instruct the Docker daemon.

- A command line interface (CLI) client [docker](#).

The CLI uses [Docker APIs](#) to control or interact with the Docker daemon through scripting or direct CLI commands. Many other Docker applications use the underlying API and CLI. The daemon creates and manages Docker objects, such as images, containers, networks, and volumes. Figure 8 shows the structure of Docker. We run commands from the Terminal and Docker daemon will get registry from Server and get data to store into the containers.

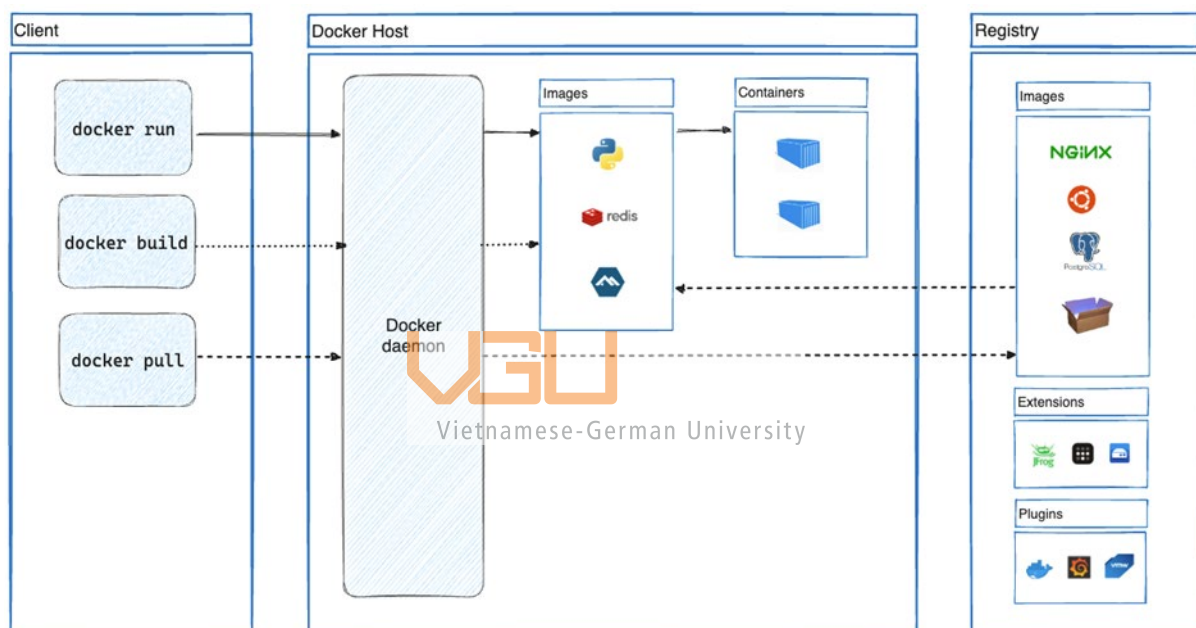


Figure 7: Docker Structure

## Docker Compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, we use a YAML file to configure our application's services. Then, with a single command, we create and start all the services from our configuration.

Compose works in all environments, production, staging, development, testing, as well as CI workflows. It also has commands for managing the whole lifecycle of our application:

- Start, stop, and rebuild services.
- View the status of running services.
- Stream the log output of running services.

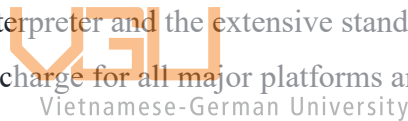
- Run a one-off command on a service.

The key features of Compose that make it effective are:

- Have multiple isolated environments on a single host
- Preserve volume data when containers are created
- Only recreate containers that have changed
- Support variables and moving a composition between environments

### 2.1.7 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.



Python has a built-in webserver provided by its standard library, which can be called for simple client-server communication. The *http.server* and *socketserver* are the two main functions used to create a web server. The port number can be defined manually in the program which is used to access the webserver. This web server is functional for many types of file formats, but it is not fully featured. Simple static HTML files can easily be parsed and served by responding with the required response code.

### 2.1.8 Django

Django is a high-level Python web framework that enables the rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so we can focus on writing our app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many free and paid-for support options.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of do not repeat ourselves. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update, and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Instagram, Mozilla, [Disqus](#), Bitbucket, Nextdoor, and Clubhouse.

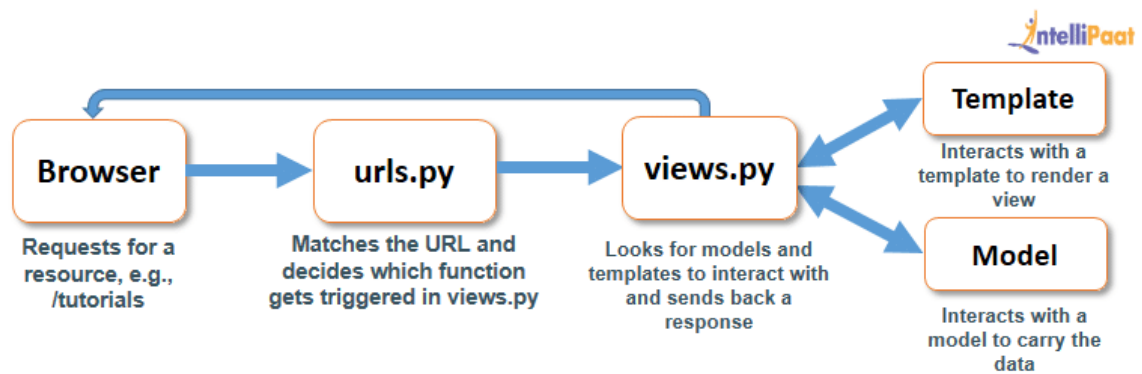



  
 Figure 8: Django components
   
 Vietnamese-German University

## 2.1.9 ReactJS

React is a popular JavaScript library used for web development. React.js is a front-end JavaScript framework developed by Facebook. To build composable user interfaces predictably and efficiently using declarative code, we use React. It is an open-source and component-based framework responsible for creating the application's view layer.

- ReactJS follows the Model View Controller (MVC) architecture, and the view layer is accountable for handling mobile and web apps.
- React is famous for building single-page applications and mobile apps.

Today's many large-scale companies (Netflix, and Instagram, to name a few) also use React JS. There are many advantages of using this framework over other frameworks, and it

is ranked among the top 10 programming languages for the last few years under various language ranking indices.

## 2.2 EdX platform users

Throughout the platform's evolution, various educational institutions and schools have proceeded to implement it to educate and evaluate the effectiveness of programs through projects. Yola is integrating edX into a program called YOLA Hybrid - the YOLA SMART Learning platform for integrated online and offline courses.

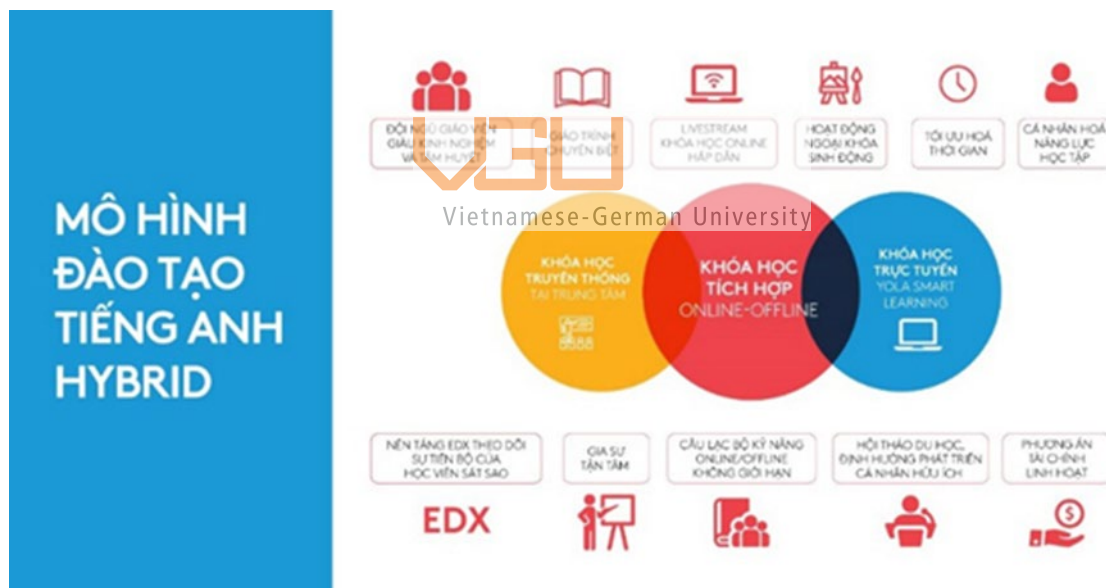


Figure 9: Yola program use edX

The screenshot displays the YOLA LMS interface. On the left is a navigation menu with a 'Vocabulary' tab and a list of weeks, including 'Week 1R - Reading (Skill Lesson 1)' which is currently selected. The main content area shows the breadcrumb 'Week 1R - Reading (Skill Lesson 1) > Homework > Strategies'. Below this is a navigation bar with 'Previous' and 'Next' buttons. The main heading is 'Strategies', followed by a blue bar for 'IELTS Reading - Matching Headings'. The 'DESCRIPTION' section lists targeted skills: identifying the main idea and differentiating it from supporting details. The 'ACTION PLAN' consists of five steps: 1. Check instructions and examples; 2. Underline key words in headings; 3. Summarize the main idea from the first paragraph; 4. Choose the best heading; 5. Avoid headings that are not supported by the text.

Figure 10: YOLA with edX

## Chapter 3: Methodology

Vietnamese-German University

In the Methodology, there are 3 parts. The first part is the Server. Server includes Ubuntu, Python, Docker, Django, CMS, and LMS. The second part is Instructors. We will set up the library, the test, and the courses. The last part is the Learners. We will set up the features for learner view.

### 3.1 Server

To build Server, we will need install following tools: Ubuntu, Python with Tutor, Docker, Django, Content Management System (CMS), Learning Management System (LMS), and create admin account for administration.

#### 3.1.1 Ubuntu

The OpenedX platform requirements run on Ubuntu. All the applications needed are supported in this OS. We use the installer to install. The installer is designed to be easy to use

and have sensible defaults so for a first install you can mostly just accept the defaults for the most straightforward install. These are our steps to install the Ubuntu.

- Choose your language.
- Update the installer (if offered)
- Select your keyboard layout.
- Do not configure networking (the installer attempts to configure wired network interfaces via DHCP, but you can continue without networking if this fails)
- Do not configure a proxy or custom mirror unless you have to in your network.
- For storage, leave “use an entire disk” checked, and choose a disk to install to, then select “Done” on the configuration screen and confirm the install.
- Enter a username, hostname, and password.
- On the SSH and snap screens, select “Done”.
- You will now see log messages as the install is completed.
- Select restart when this is complete, and log in using the username and password provided.



Vietnamese-German University

### 3.1.2 Python

The OpenedX platform currently have 3 supported installation methods: Native, Devstack, and Tutor. We will be using Tutor for this project. Tutor is a community-supported Docker-based environment suited for both production and development. The goal of Tutor is to make it easy to deploy, customize, upgrade and scale Open edX. Tutor is reliable, fast, extensible, and it is already used to deploy hundreds of Open edX platforms around the world. Tutor using Python language to build OpenedX platform.

There is QuickStart with a 1-click install option which can automatically clone the repository of Tutor from GitHub and install all requirements that are needed to run the platform or we can check by steps:

- `sudo apt install python3 python3-pip libyaml-dev`
- `pip install "tutor[full]==16.1.2"`

### 3.1.3 Docker and Docker Compose Installation

The Docker installation package provided from the official Ubuntu repository may be out of date. We will install Docker from the official Docker repository to ensure we get the most recent version. To do so, we will create a new package source, import the Docker GPG key to guarantee the downloads are legitimate, and then install the package. In Terminal, we run the following commands sequentially.

For Docker installation steps:

- `sudo apt update`
- `sudo apt install apt-transport-https ca-certificates curl software-properties-common`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"`
- `sudo apt install docker-ce`
- `sudo usermod -aG docker ${USER}`
- `su - ${USER}`

For Docker Compose installation steps:

- `sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
- `sudo chmod +x /usr/local/bin/docker-compose`

### 3.1.4 Django

Django is used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework and can deliver content in almost any format (including HTML, RSS feeds, JSON, and XML). To install the Django, we were recommended to install an official release with pip. We need to update pip if it is outdated. The steps to install are:

- Create virtual environments with `python -m venv openedx-env` and `source openedx-env/bin/activate`
- After you have created and activated a virtual environment, we install by enter command `python -m pip install Django`

### 3.1.5 CMS

CMS is Content Management System software. The software is used to create, manage, and modify content on a website. The CMS software allows users without technical knowledge to create courses.

Depending on our deployment target, run one of:

```
tutor local launch # for local installations.
tutor dev launch   # for local development installations
tutor k8s launch   # for Kubernetes installation.
```

We will answer a few questions about the Configuration of our Open edX platform which is stored in a single `config.yml` file. To view the content of this file, run:

```
cat "$(tutor config printroot)/config.yml"
```

By default, this file contains only the required configuration parameters for running the platform. Optional configuration parameters may also be specified to modify the default behavior. After the launch command is done

A full, production-ready Open edX platform ([Palm](#) release) is run with `docker-compose`. The whole procedure should require less than 10 minutes, on a server with good bandwidth. Note that our host environment will not be affected in any way, since everything runs inside docker containers. Root access is not even necessary.

### 3.1.6 LMS

LMS is Learning Management System software. The software is a digital platform to make learning more efficient, manage users, and organize data. An LMS facilitates access to learning materials, allows administrators to oversee user activity and course details, and

automates tasks like grading and reporting to streamline training processes. In other words, an LMS enables the comprehensive management of the entire learning process.

Let us break down each component of the acronym and the very concept of an LMS:

- L — Learning. This type of software is designed for convenient learning anytime, anywhere. Learners can access an in-house knowledge base, take assessments, and complete practical assignments.
- M — Management. Unlike file sharing services, a learning platform is not just a heap of files. It enables admins to manage user data and enrollments in courses, organize content into comprehensive training programs, and control how learners progress in the training they are taking.
- S — System. It is a computer system that processes various types of data and puts them all in order. It collects statistics, runs reports, and automates grading, streamlining the entire training process.

### 3.1.7 Create admin account

Out of the box, Tutor does not create any user for us. We will want to create a user ourselves with staff and administrator privileges to access the studio. There is a simple command for that.

```
tutor local/dev do createuser --staff --superuser  
yourusername user@email.com
```

We will be asked to set the user password interactively.

## 3.2 Instructor

### 3.2.1 XBlocks

Course creators must be able to combine components from many sources in order to produce rich, interesting online courses. This is made feasible via edX's component design, XBlock. Courses are constructed in a hierarchical manner using XBlocks. XBlocks, like HTML elements, can represent as little as a paragraph of text, a video, or a multiple-choice input box, or as much as a section, a chapter, or an entire course.

XBlocks are not just for providing courses. A comprehensive educational ecosystem will include a variety of online apps, all of which will require access to course material and data. The structure and APIs required to construct components for usage in any of these apps are provided by XBlocks.

### 3.2.2 LTI

We may have discovered or developed an external learning application that we would like to integrate into our online course. Alternatively, we may have a digital copy of our textbook in a format other than PDF. Using a Learning Tools Interoperability (LTI) component, we may integrate external learning programs or textbooks into Studio. The LTI component is based on the IMS Global Learning Tools Interoperability standard version 1.1.1.

We can use an LTI component in several ways.

- Add external LTI content that is displayed only, such as textbook content that does not require a student response.
- Add external LTI content that requires a student response. An external provider will grade student responses.
- Use the component as a placeholder for syncing with an external grading system.

### 3.2.3 API

Open edX includes a suite of APIs that allow us to build applications that interact with the core platform.

Open edX APIs use Representational State Transfer (ReST) design principles and support JavaScript Object Notation (JSON) data-interchange format.

Our APIs use OAuth 2.0 for authentication. OAuth 2.0 is an open standard used by many systems that require secure user authentication. See the OAuth 2.0 Standard for more information.

The Open edX Platform ReST APIs are a rapidly growing and evolving set of capabilities that enable us to build web, desktop, and mobile applications that work with our Open edX instance.

Currently, the Open edX Platform includes the following APIs: Enrollment API, User API, and Data Analytics API.

### 3.2.4 Create course

To create and run a course, you can either create an entirely new course in Studio, or you can re-run an existing course. You can also export and import courses as XML files. You can do this when you need to back up a course or edit the course in XML. To create a course, follow these steps.

1. Sign in to Studio.
2. Select **New Course**.
3. Enter the following information about your course.
  - For **Course Name**, enter the title of your course. For example, the name might be “Sets, Maps, and Symmetry Groups”. Use title capitalization for the course title, and normal spacing and punctuation.
  - For **Organization**, you enter the identifier for your organization. Do not include spaces or special characters.
  - For **Course Number**, you enter both a subject abbreviation and a number. For example, for public health course number 207, enter PH207. For math course 101x, enter Math101x. Do not include spaces or special characters in the course number.
  - For **Course Run**, you enter the term in which your course will run. For example, enter 2014SOND or T2\_2014. Do not include spaces or special characters.

The value that you enter for the run does not affect the course start date that you define for the course. The organization, course number, and course run values that you enter are used to create the URL for your course. Because URLs do not have an unlimited length, the total number of characters that you enter for these values must be 65 or fewer.

#### 4. Select **Create**.

The Course Outline page opens. You can start developing course content.

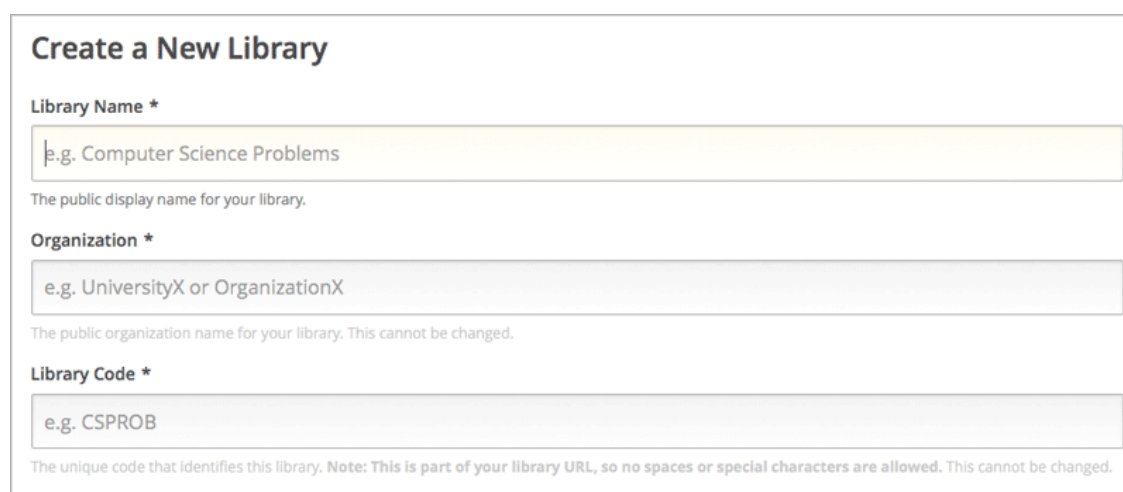
After you create a course, the course opens in Studio automatically and you can begin editing. We can Set Start and End Dates and Times for courses

### 3.2.5 Create library

In Studio, we can create a library to build a pool of components for use in randomized assignments in our courses. We can add Text components, problems, and video components to a library. After creating a library and adding components to it, if we have enabled content libraries in our course, we can use these library components in randomized assignments in our course. We do this by adding a randomized content block to a course unit and specifying the name of the library from which the randomized content is to be drawn. We also specify the number and type of problems that each learner is assigned. Libraries have separate users and levels of access from courses. Initially, only the person who created the library has access. The author of the library can add other users to the library.

To create a new library, follow these steps.

1. Log in to Studio.
2. Select **New Library**.
3. Enter the required information for your new library, then select **Create**.



**Create a New Library**

**Library Name \***  
e.g. Computer Science Problems  
The public display name for your library.

**Organization \***  
e.g. UniversityX or OrganizationX  
The public organization name for your library. This cannot be changed.

**Library Code \***  
e.g. CSPROB  
The unique code that identifies this library. Note: This is part of your library URL, so no spaces or special characters are allowed. This cannot be changed.

Figure 11: Create library

- For **Library Name**, enter the public display name for your library. Choose a meaningful name that will help you and other course team members to identify the library. For example, “Level 200 Math Problems”. When you add a randomized content block to a course unit, you use the library name to specify this library as a source for the randomized assignment.
- For **Organization**, enter the identifier for your university. For example, enter HarvardX or MITx. Do not include spaces or special characters.
- For **the Library Code**, enter an identifier for your library that is unique within your organization. This code becomes part of the URL for your library, so do not include spaces or special characters in the code.

#### 4. Select **Create**.

After we create a library, we are automatically assigned an Admin role for the library.

### 3.2.6 Create Problem

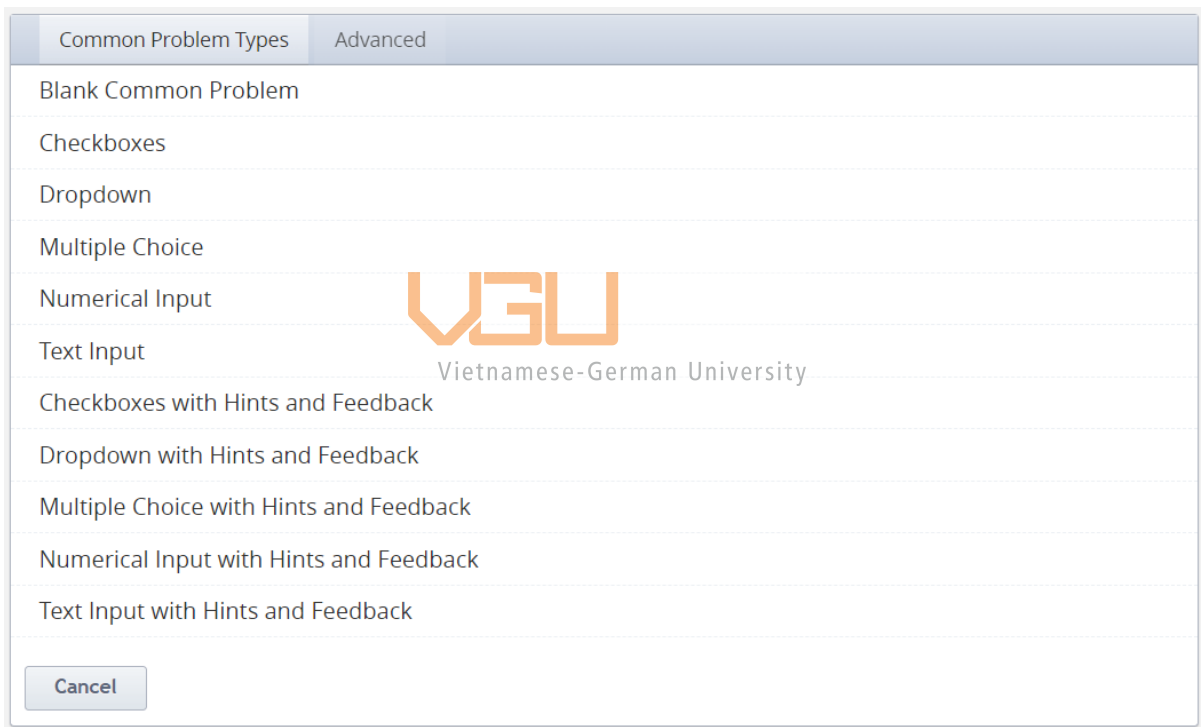
When you select **Problem**, Studio adds a blank problem to the unit and brings you to the problem editor. From here, you can select 1 of 5 **simple problem types** or select Advanced problem types for a list of **advanced problem types**.

- For simple problem types, the simple editor opens. In this editor, you can quickly create problems with question-and-answer formats. Options for scoring, hints, feedback and more can be entered.
- For advanced problem types, the advanced editor opens. In this editor you use open learning XML (OLX) elements and attributes to identify the elements of the problem. Options for scoring, feedback and more can be entered.
- For open response assessment problem types, you define the problem elements and options by using a graphical user interface. For more information, see [Create an Open Response Assessment Assignment](#).
- For drag and drop problem types, you build an interactive assessment in a customized interface in which you define areas that learners can drag into target zones on a background image. For more information, see [Creating a Drag and Drop Problem](#).

You can switch from the simple editor to the advanced editor at any time by selecting the **Switch to advanced editor** from the simple editor's settings.

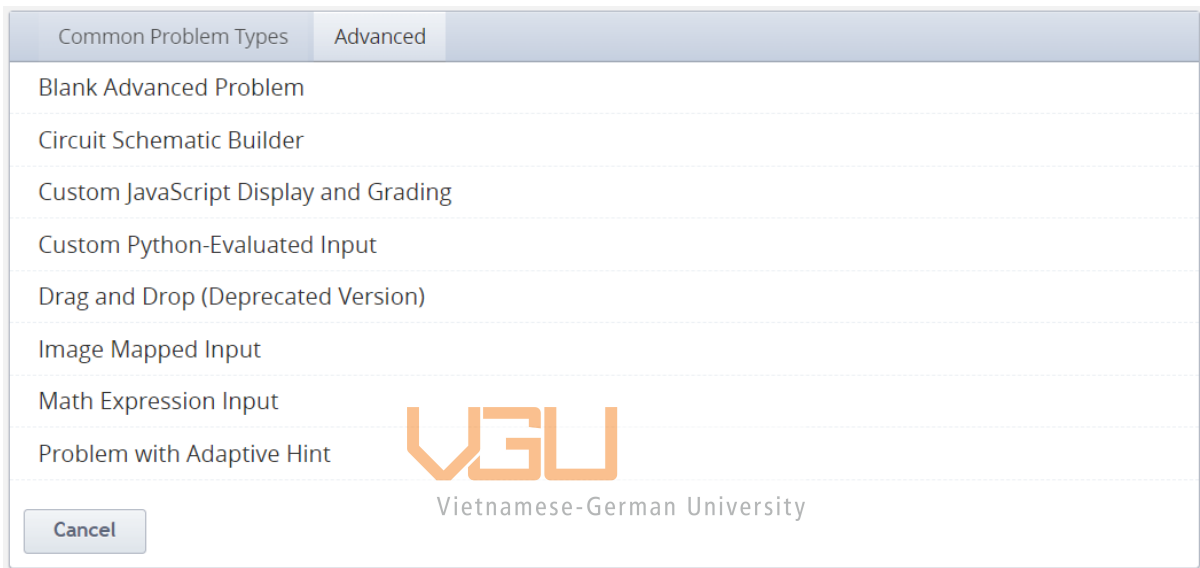
When you select one of the **simple problem types**, you will be directed to the simple editor.

- Single Select
- Multi-select
- Dropdown
- Numerical Input
- Text Input



If the simple editor cannot fulfill your needs, you might turn your attention to the Advanced Editor. This editor will allow you to directly edit the open learning XML (OLX) of your problem. If the simple editor cannot fulfill your needs, you might turn your attention to the Advanced Editor. This editor will allow you to directly edit the open learning XML (OLX) of your problem. Since the Advanced Editor allows you to edit the problem directly using the OLX, there are many more ways to write a problem. Below are several features the Advanced Editor is capable of:

- Blank Advance Problem
- Custom JavaScript Display and Grading
- Custom Python-evaluated Input (Write Your Own Grader)
- Math Expression Input Problems
- Circuit Schematic Builder Problem
- Circuit Schematic Builder Problem
- Problem with Adaptive Hint



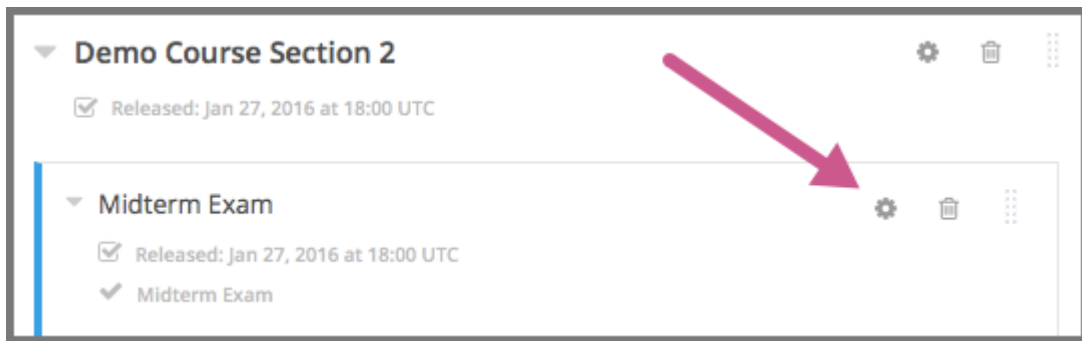
### 3.2.7 Create Exam

To create a proctored exam or a practice proctored exam, follow these steps.

1. Add and develop a subsection as you would any other subsection.

For a practice proctored exam, edX recommends that you include only a few problems in this subsection. Or, you could add a unit with a text component that provides information about the exam.

2. Select the **Configure** icon for the subsection.



The **Settings** dialog box opens to the **Basic** tab.

3. In the **Grading** section, set the assignment type and due date for the subsection.
4. Select the **Advanced** tab, and then locate the **Set as a Special Exam** section.
5. Select the proctored exam type.
  - For a proctored exam, select **Proctored**.
  - For a practice proctored exam, select **Practice Proctored**.
6. In the **Time Allotted** field, enter the length of time that you want to allow for the exam as HH:MM, where HH is hours and MM is minutes.
7. (optional) For a proctored exam, in the **Review Rules** field, enter any additions or exceptions to the default rules for proctored exams. For more information, see Specify Exam Rules and Exceptions.
8. Select **Save**.

The rules for proctoring exams vary according to the proctoring service that you or your organization has chosen. However, the online proctoring rules that this guide lists are common to many proctoring services.

You can also provide custom rules for a particular exam. For example, you might allow learners to use a calculator, or to have one sheet of blank paper. These rules are visible to reviewers and learners at the start of each proctoring exam.

To specify custom proctored exam rules, follow these steps.

1. In Studio, open your course outline and select the subsection for the exam.
2. Select the **Configure** icon to open the **Settings** dialog box.
3. Select the **Advanced** tab, and then locate the **Set as a Special Exam** section.

4. In the **Review Rules** field, enter any additional rules. For example, you might enter the following rule.

Allow one blank sheet of paper and a pencil or pen. Allow a hand-held calculator on the desk.

5. Select **Save**.

When a proctored exam opens, by default, learners are required to take the exam as a proctored exam. If you want to allow learners to take proctored exams without proctoring, you can change a setting on the **Proctored Exam Settings** page in Studio.

To enable the option for learners to opt out of proctored exams for a course, follow these steps.

1. In Studio, select **Settings**, then select **Proctored Exam Settings**.
2. Locate the **Allow Opting Out of Proctored Exams** policy key. The default value is No, which requires Verified and Master's learners to take proctored exams with proctoring.
3. Change the value of the setting to Yes.
4. Select **Submit**.



Vietnamese-German University

## 3.3 Learner

After a fresh installation, the LMS platform will not have a single course. To import the Open edX demo course, we will run:

```
tutor local/dev do importdemocourse
```

To get a glimpse of the possibilities of Open edX

There are features for the learner include explore their dashboard, enroll in a Course, check the progress of the course, get the course certificate, and participate into discussion course

### 3.3.1 Enroll a Course

To enroll in a course, follow these steps.

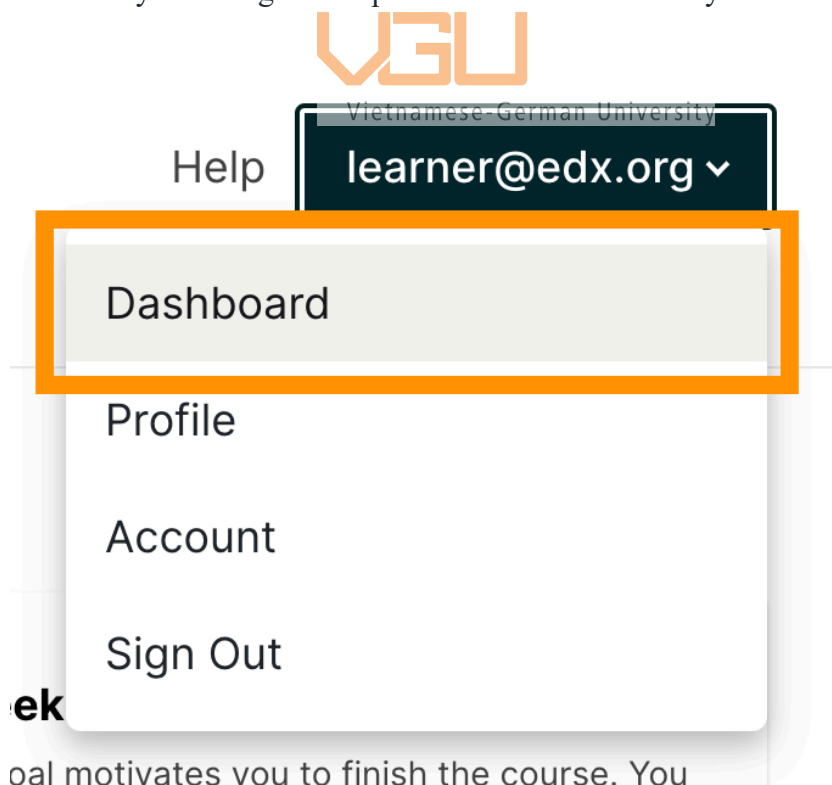
1. In the course catalog on the main website, select the course that you want to take.
2. On the About page for the course, select **Enroll Now**.

For some courses, different enrollment options might be available, such as a free audit track or a paid certificate track. If the course offers different enrollment options, select an option.

As soon as you enroll in a course, it is added to the list of courses on the **Courses** page of your dashboard. You can start working in the course if the course start date has passed. To open the course, locate the course listing on your dashboard and select **View Course**.

### 3.3.2 Explore Dashboard

Your dashboard lists all of the courses that you are enrolled in. When you create an account or enroll in a course, you automatically go to your dashboard. You can also access your dashboard by selecting the dropdown menu icon next to your username at the top of any page.



goal motivates you to finish the course. You

You can find information about your courses and change course-specific email subscription settings on your dashboard.

The dashboard has three main pages.

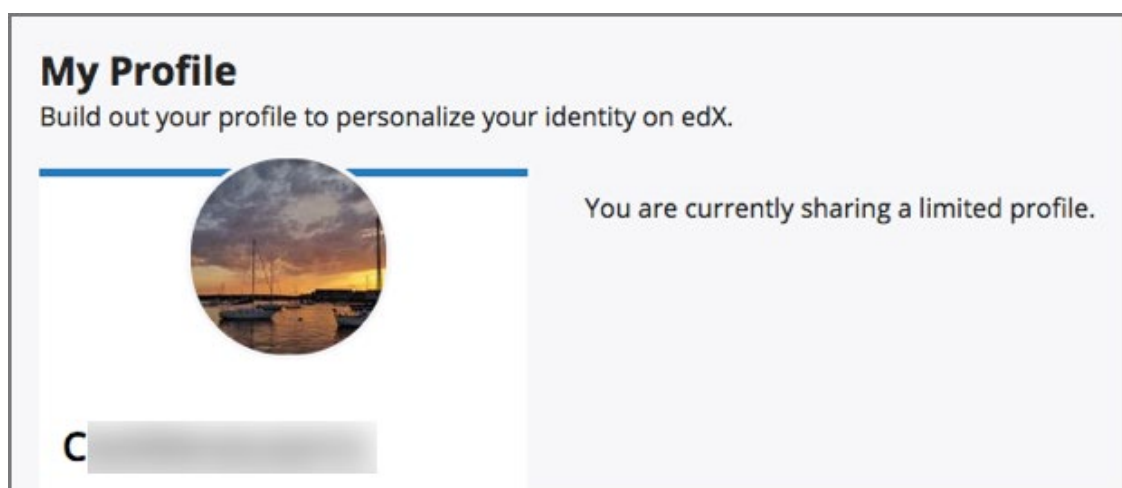
- The **Courses** page provides information about your courses and allows you to change course settings. By default, the dashboard opens to the **Courses** page.

The **Courses** page on the dashboard now includes two main categories of filters with subcategories in each main category. **Course Status** includes ‘In progress,’ ‘Not Started,’ ‘Done,’ ‘Not Enrolled,’ and ‘Upgraded.’ These different filters allow you to filter all of the courses based on the course status. **Sort** includes ‘Last enrolled’ and ‘Title (A-Z).’ These filters allow you to filter the course alphabetically or when you were last enrolled in the course.

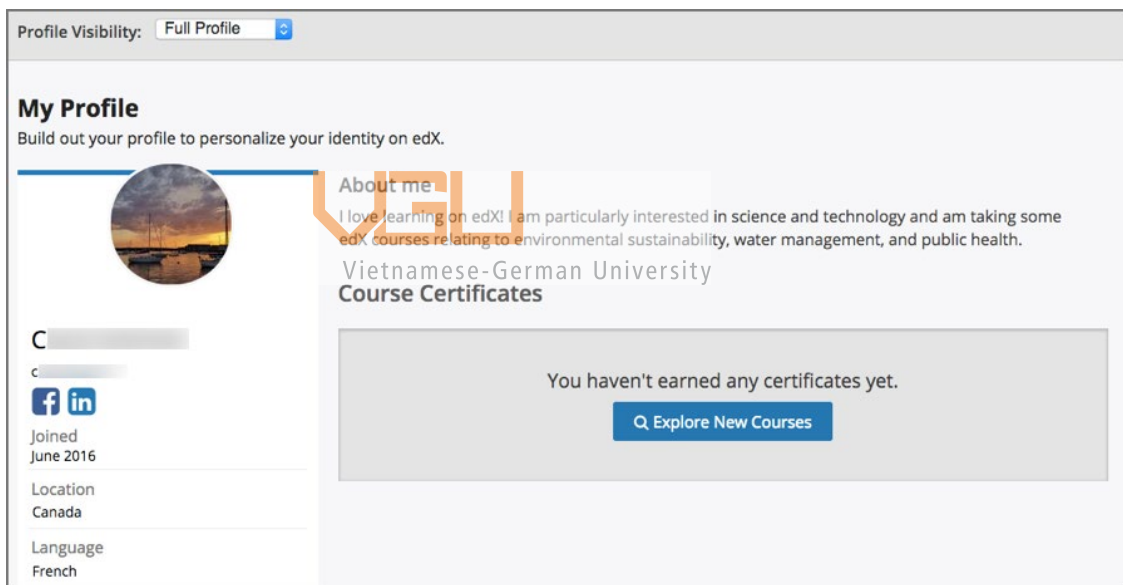
- The **Programs** page lists any programs, such as XSeries or MicroMasters programs, that edX offers for courses that you are enrolled in. Programs appear on this page if you are enrolled in any course that is part of that program. This page also shows how many courses in the program you have completed, how many are in progress, and the number of remaining courses you have left in the program. For more information, see Explore edX Programs.
- The **Discover New** page take you to the edX website and will allow you to search for any courses on the edX platform that interest you.

Learner can also add profile information. The personal profile allows you to optionally share information about yourself with other participants in the courses that you take. If you choose to add profile information, course teams and other learners in your courses can view the information that you share.

- You can share either a limited profile or a full profile. A limited profile shares only your username. You can also add a profile picture if you are over 13 years of age.



- A full profile includes the following details.
  - Your full name
  - Your username
  - Your profile image (optional)
  - The date that you joined edX
  - Your language (optional)
  - Your location (optional)
  - Information about you (optional)
  - Links to any course certificates that you have earned
  - Links to your social media accounts (optional)



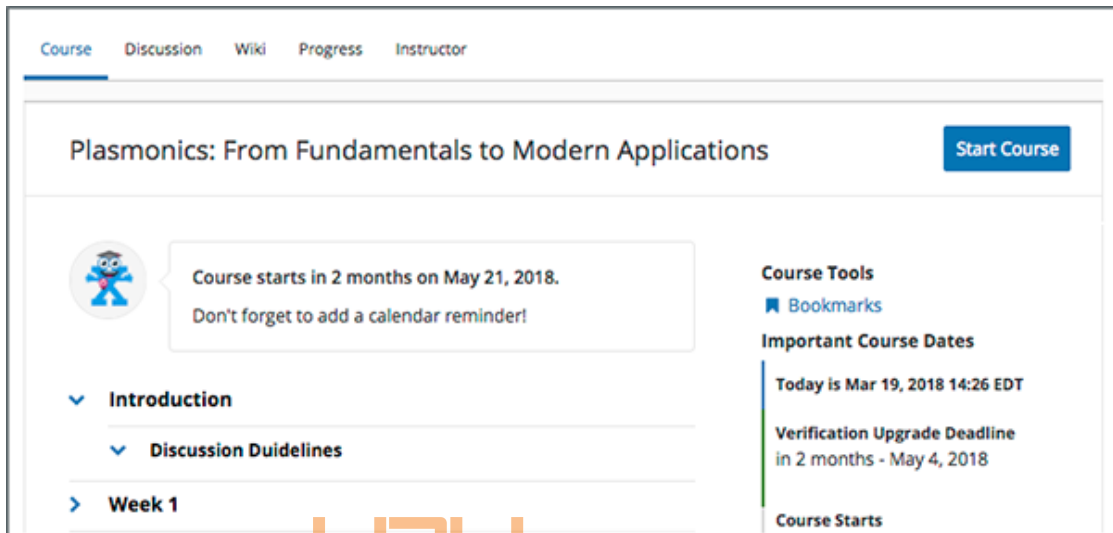
### 3.3.3 Check progress

When you are working in a course, green check marks that show your progress are visible in two places.

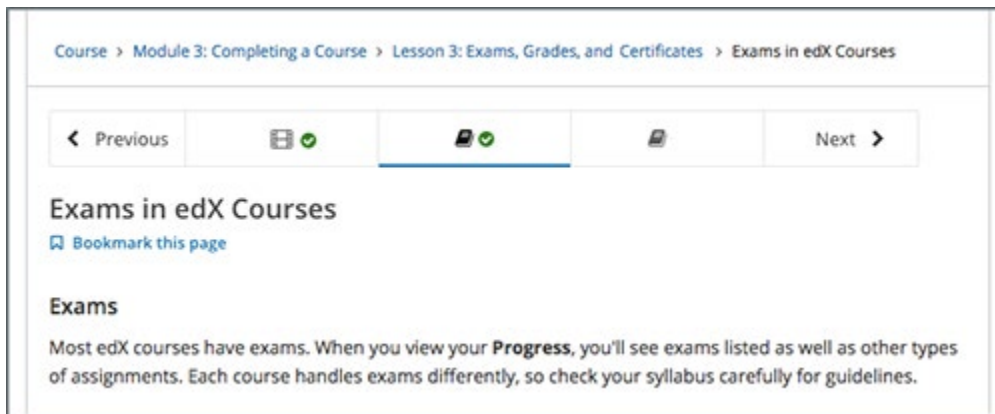
- The course outline on the **Course** page.
- The navigation bar at the top of the page.

These check marks appear when you complete all of the content in a part of the course. The part might be called a unit, section, module, week, lesson, or other part, depending on the terms your course uses.

For example, in the following image, the learner has completed all of the content in the “Introduction” part of the course, as well as some of the content in the “Module 1: Experiencing Course Content” part.



In the following image, the learner has completed the first two units in the “Lesson 3: Exams, Grades, and Certificates” lesson.



To complete content, you must complete all of the following actions.

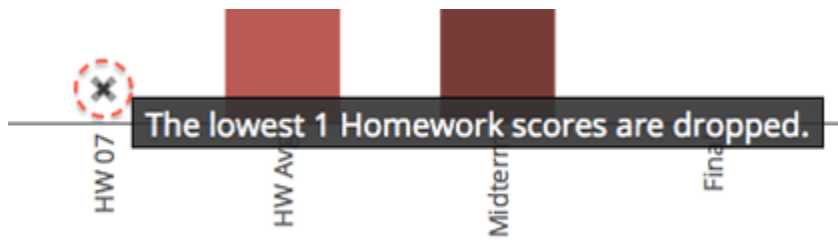
- View all of each video in the unit.
- Submit answers for all of the problems in the unit.
- View all HTML content in the unit for at least five seconds.

There is also have a Progress Page to see your scores on individual problems represented in both a chart and a list, as well as your current overall course grade. On the **Progress** page, a chart shows how you have performed for each graded assignment in the course.

Assignments are listed along the horizontal axis, and the percentage score that you received in each lesson is read from the vertical axis.



Markers on the vertical axis reflect the grading scale for the course. For example, if the passing percentage score for the course is 60%, there is a marker on the vertical axis at 60% so that you can compare your total score with the required passing score. If the course awards a grade of B for scores of 50% and higher, and a grade of A for scores 75% and higher, then there are markers on the vertical axis at 50% and 75%, labelled “B” and “A” respectively. The bar for each assignment reflects your total score for all the problems in that assignment. For individual problem scores, see Grading Details below the chart. For each of the assignment types, an “average” bar shows the current average of scores for assignments of that type. This average is recalculated as you progress through the course and complete more assignments. Some courses allow some number of graded assignments to be automatically dropped from your final score. For example, out of 8 quizzes, a course might have the lowest scoring quiz excluded from all learners’ final scores. A dropped assignment is indicated in the chart by an x above the horizontal axis.



The **Progress** page includes your scores for individual problems within an assignment. Below the grading chart is a listing of course sections in the order that they occur in the course.

For each problem in a graded assignment, the points that you earned out of the possible points is shown with the label **Problem Scores**. Scores for ungraded problems are shown with the label **Practice Scores**. Sections that do not have any problems are labelled as having no problem scores.

**Lesson 3: The Minor Pentatonic Scale**

**Lesson 3 Overview**  
 No problem scores in this section

**Spotlight: Advice for Studying Music**  
 No problem scores in this section

Vietnamese-German University

**Review from Lesson 2**  
 No problem scores in this section

**Major 6th and Major 7th Intervals**  
 Practice Scores: 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0

**Building and Singing the Minor Pentatonic Scale**  
 Practice Scores: 0/0 0/0

**Berklee Student Performance: "Work In Progress"**  
 No problem scores in this section

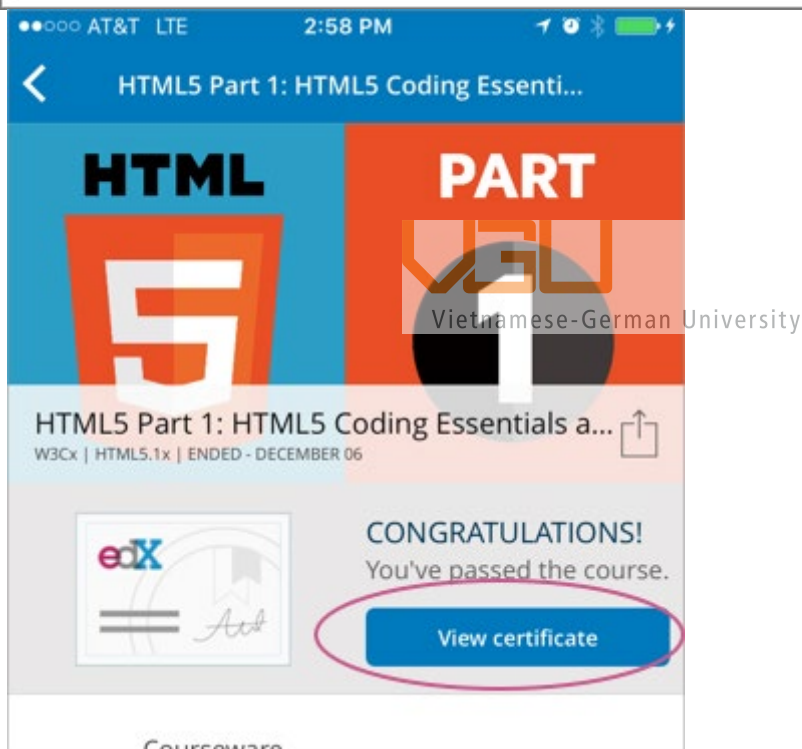
**Lesson 3 Quiz (0/15)**  
 Quizzes due Feb 15, 2016 at 23:00 UTC  
 Problem Scores: 0/15

**Lesson 3 Review and Assignment Overview (0/8)**  
 Peer Reviews due Feb 23, 2016 at 23:00 UTC  
 Problem Scores: 0/8

### 3.3.4 Get Certificate

Some courses issue certificates, which you can access from your dashboard.

The following examples show the **View Certificate** option on a learner dashboard on the edx.org website and on the mobile app.



To open a web certificate in your browser, select **View Certificate**. You can then share or print your web certificate.

## Chapter 4: Results and Discussions

### 4.1 Server

#### 4.1.1 Tutor

Tutor comes with a plugin system that allows anyone to customize the deployment of an Open edX platform very easily. The vision behind this plugin system is that users should not have to fork the Tutor repository to customize their deployments. For instance, if we have created a new application that integrates with Open edX, we should not have to describe how to manually patch the platform settings, `urls.py` or `*.env.yml` files. Instead, we can create a “tutor-myapp” plugin for Tutor. This plugin will be in charge of making changes to the platform settings. Then, users will be able to use our application in three simple steps:

# 1) Install the plugin.

```
pip install tutor-myapp
```

# 2) Enable the plugin.

```
tutor plugins enable myapp
```

# 3) Reconfigure and restart the platform.

```
tutor local/dev launch
```

For simple changes, it may be extremely easy to create a Tutor plugin: even for non-technical users.

Additionally, Overhang.IO GitHub provides several plugin repositories:

- **MFE:** The plugin makes it possible to easily add micro frontend (MFE) applications on top of an Open edX platform that runs with Tutor. In addition, this plugin comes with a few MFEs which are enabled by default:

- **Authn:** This is a micro-frontend application responsible for the login,

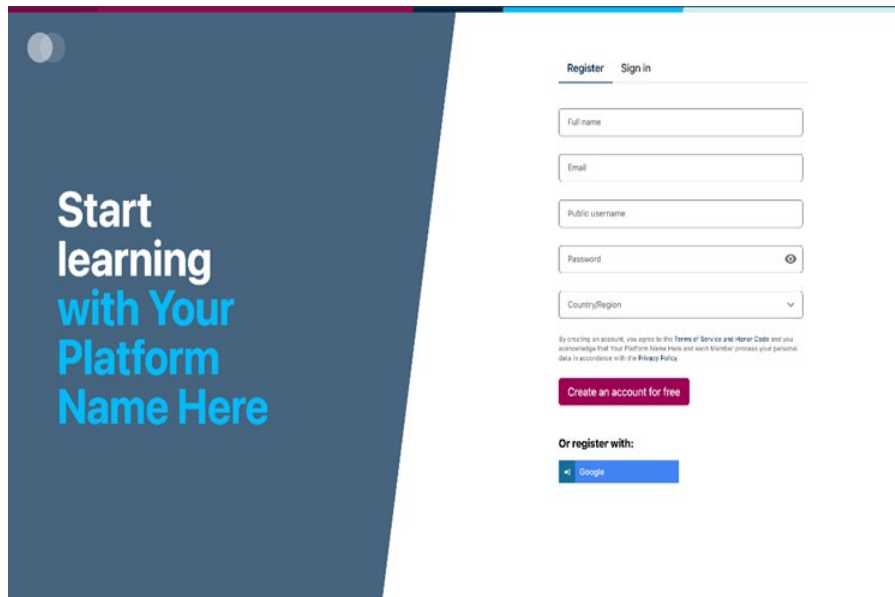


Figure 12: Authn page

registration, and password reset functionality.

- **Account:** An MFE to manage account-specific information for every LMS user. Each user's account page is available at `http(s)://{ MFE_HOST }/account`. For instance, when running locally: <https://apps.local.overhang.io/account>.

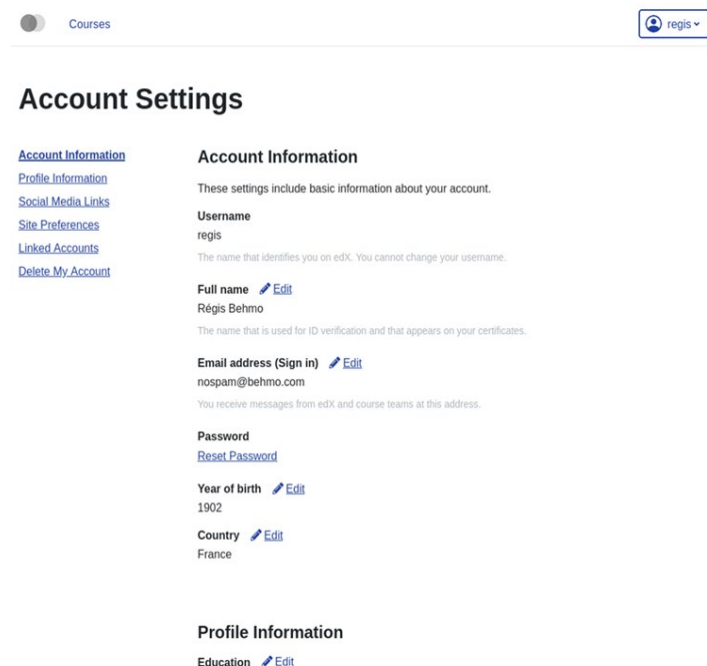


Figure 13: Account page

- **Communications:** The Communications micro-frontend exposes an interface for course teams to communicate with learners. It achieves this by allowing instructors to send out emails in bulk, either by scheduling them or on

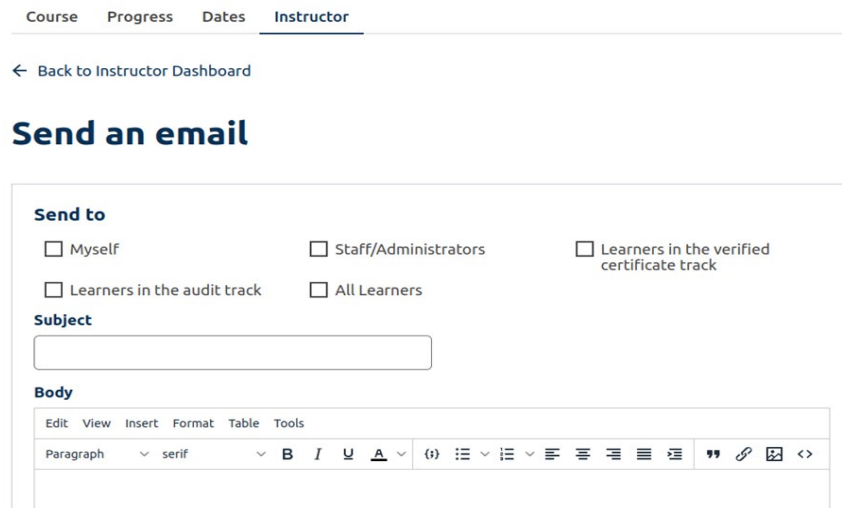


Figure 14: Communication page

demand.

- **Course Authoring:** This MFE is meant for course authors and maintainers. For a given course, it exposes a "Pages & Resources" menu in Studio where one can enable or disable a variety of features, including, for example, Wiki and Discussions. Optionally, it allows authors to replace the legacy HTML, Video, and Problem authoring tools with experimental React-based versions, as well as exposing a new proctoring interface that can be enabled if the [edx-exams](#) service is available.

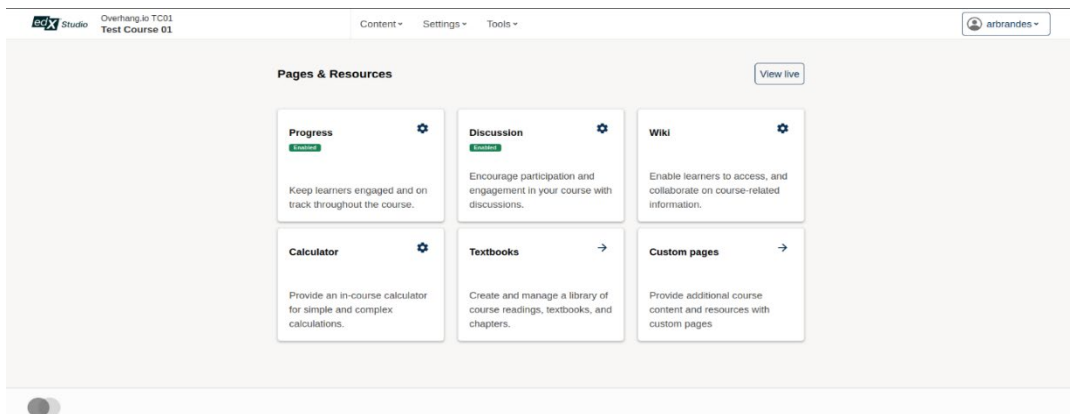


Figure 15: Course Authoring page

## ○ Discussions:

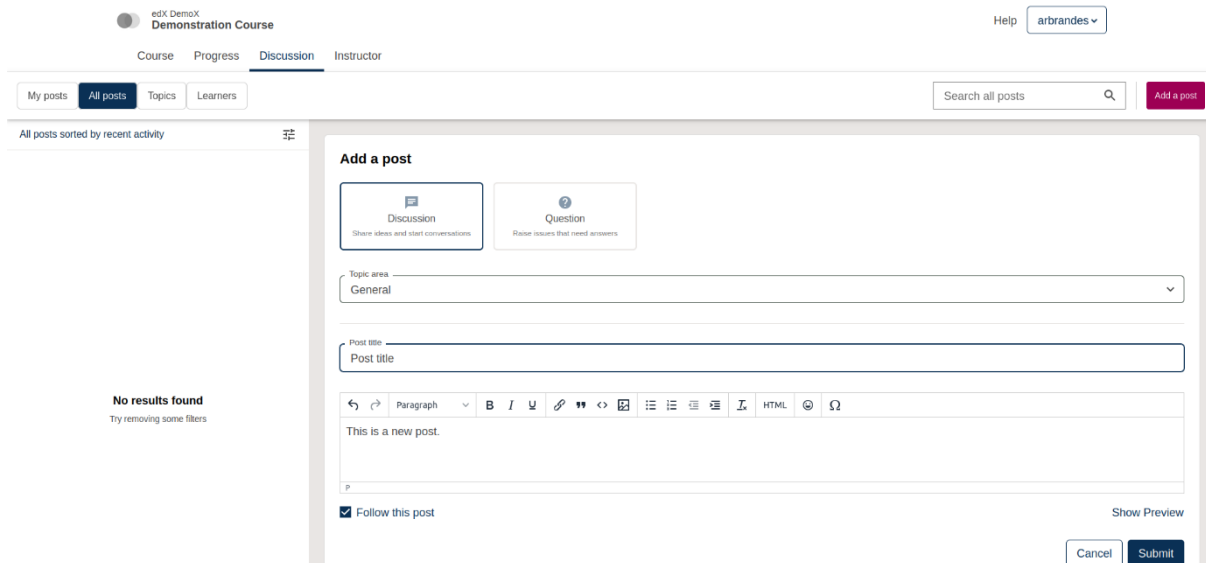


Figure 16: Discussion page

- **Gradebook:** This instructor-only MFE is for viewing individual and aggregated grade results for a course. To access this MFE, go to a course → Instructor tab → Student Admin → View gradebook. The URL should be: `http(s)://{{ MFE_HOST }}/gradebook/{{ course ID }}`. When running locally, the gradebook of the demo course is available at: [http://apps.local.overhang.io/gradebook/course-v1:edX+DemoX+Demo\\_Course](http://apps.local.overhang.io/gradebook/course-v1:edX+DemoX+Demo_Course)



[<< Back to Dashboard](#)

# Gradebook

course-v1:edX+DemoX+Demo\_Course

Grades

## Step 1: Filter the Grade Report

Edit Filters

regis ✕ 🔍

Search by username, email, or student key

## Step 2: View or Modify Individual Grades

Showing 1 of 1 total learners

Score View:

Percent ▾

Username Student Key*	Email*	Ex 01	Ex 02	Exam 01	Total
regis		<a href="#">0%</a>	<a href="#">0%</a>	<a href="#">0%</a>	0%



Figure 17: Gradebook page

Vietnamese-German University

- Learning:

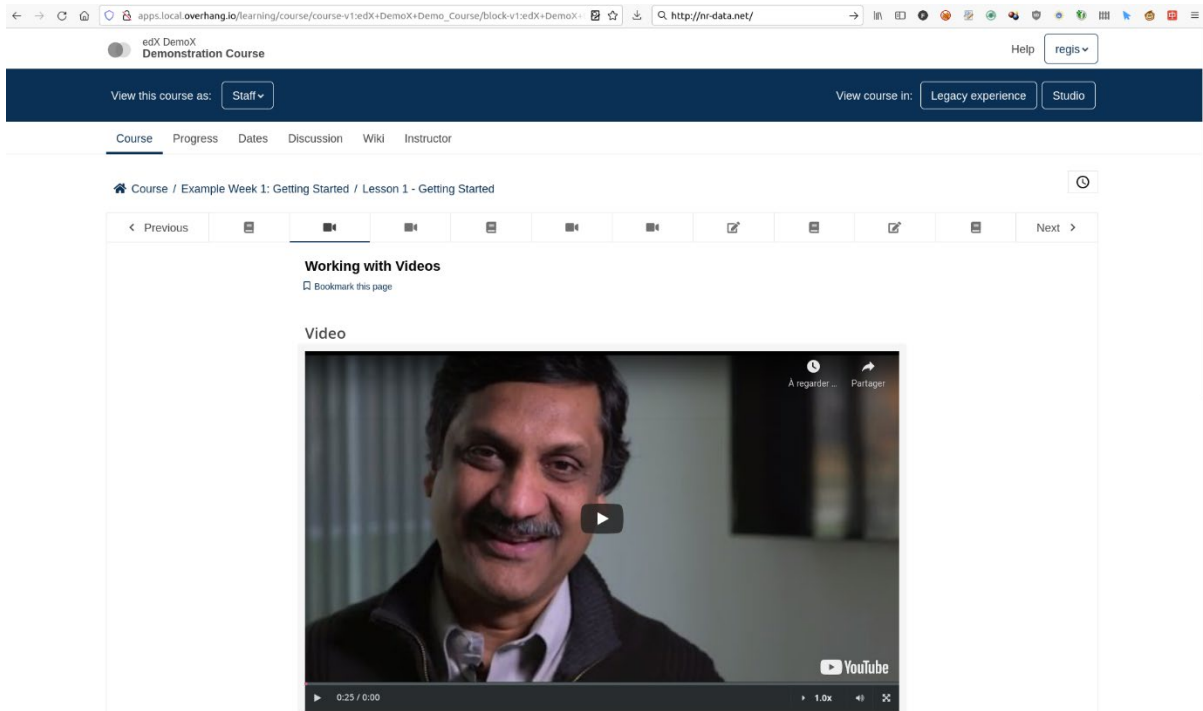


Figure 18: Learning page

- **ORA Grading:** When enabled, Open Response Assessments ("ORA") that have a staff grading step will link to this new MFE, either when clicking "Grade Available Responses" from the exercise itself, or via a link in the Instructor Dashboard. It is meant to streamline the grading process with better previews of submitted content.

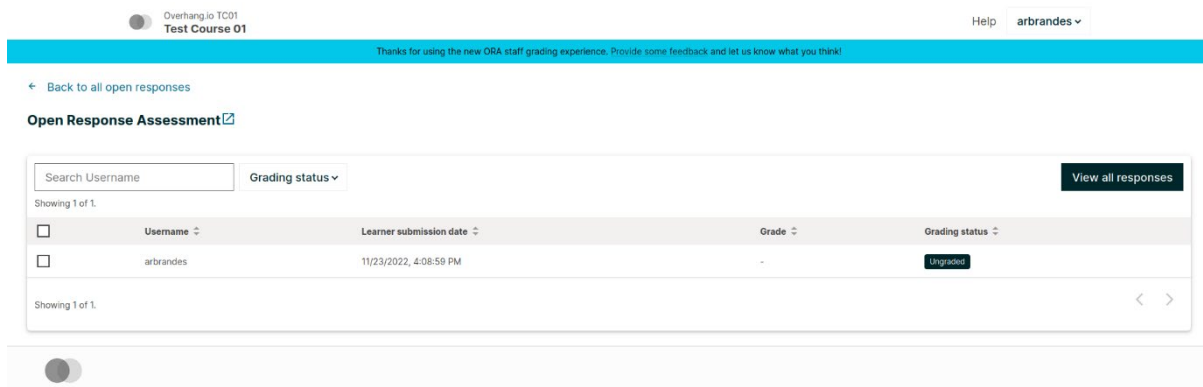


Figure 19: ORA page

- **Profile:** Edit and display user-specific profile information. The profile page of every user is visible at `http(s)://{{ MFE_HOST }}/profile/u/{{ username }}`. For instance, when running locally, the profile page of the "admin" user is:

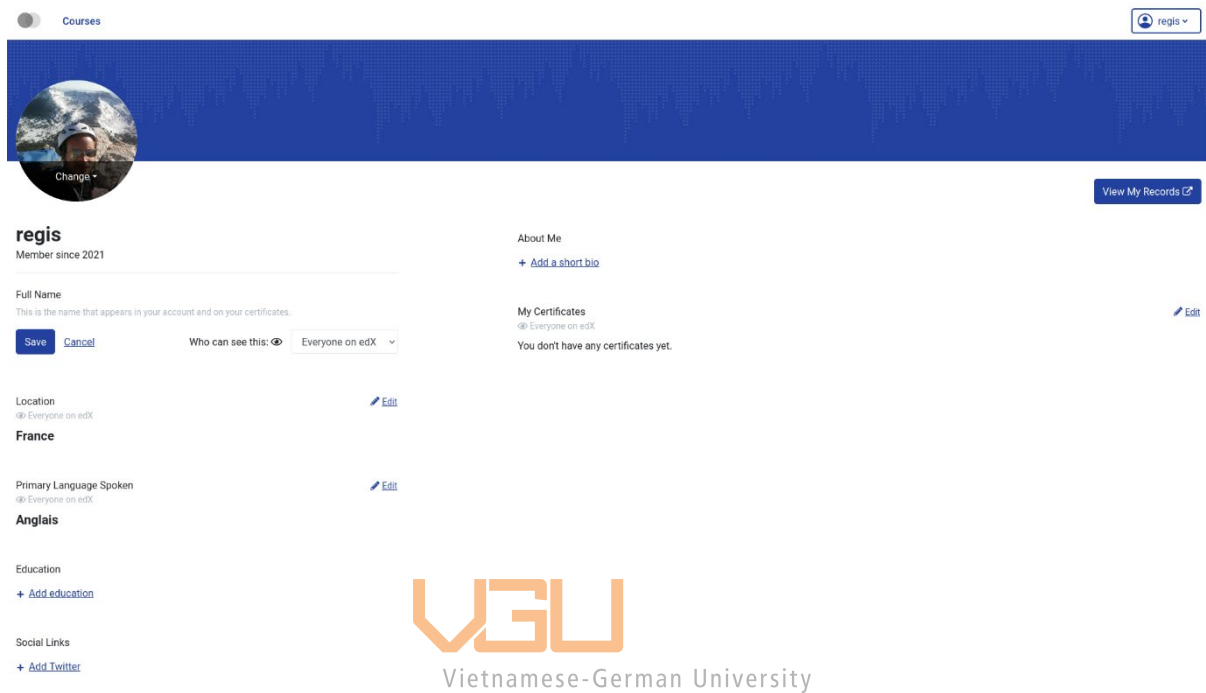


Figure 20: Profile page

<http://apps.local.overhang.io/profile/u/admin>.

- **Forum:** The plugin adds discussion forums to our Open edX platform, such that students can have conversations about the courses they are following right in our LMS.
- **Discovery:** This is a plugin for Tutor that integrates the Course Discovery application in an Open edX platform. it is useful for integration with, for example, Ecommerce or an external course catalog.
- **Xqueue:** This is a plugin for [Tutor](#) that provides the Xqueue external grading system for Open edX platforms.

- **Notes:** This is a plugin for [Tutor](#) to easily add the [Open edX note-taking app](#) to an Open edX platform. This app allows students to annotate portions of the

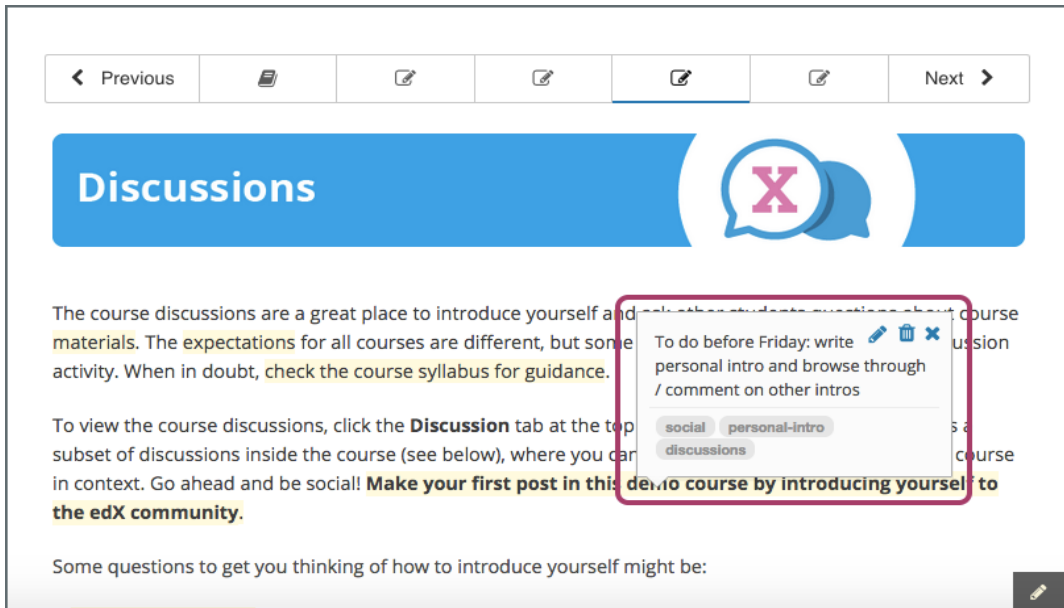


Figure 21: Notes plugin

courseware.

- **Credentials:** This is a plugin for [Tutor](#) that integrates the [Credentials](#) application in an Open edX platform. The credentials application supports course and program certificates. This plugin offers an admin panel where user can do configurations for the certificates of his course and program.

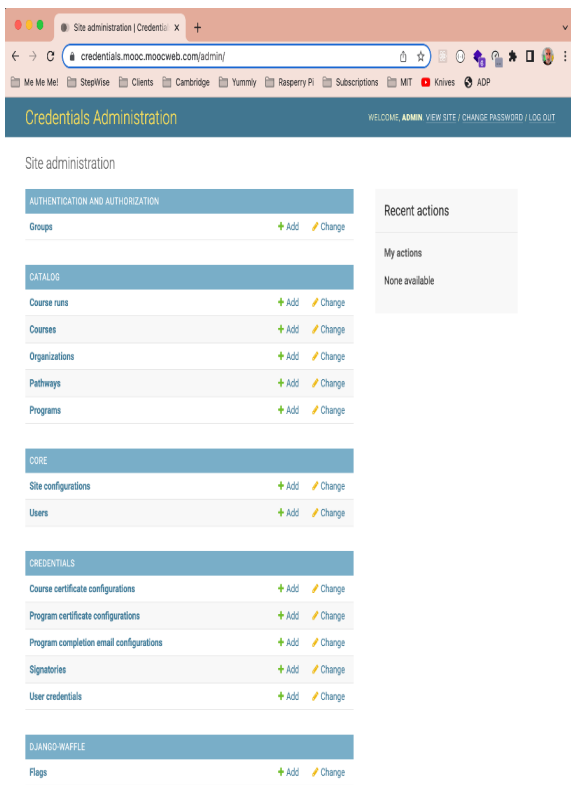
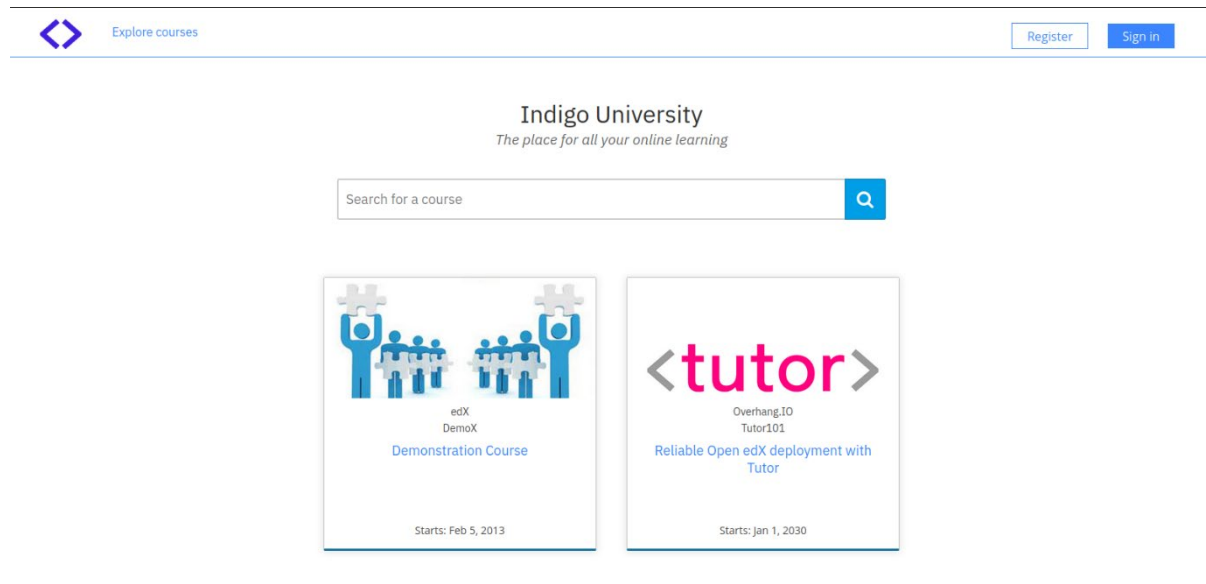


Figure 22: Credentials plugin

Note that user will have to create the course/program using the [Discovery plugin](#). Then Credentials plugin will be used for certificate configurations.

- **Indigo:** Indigo is an elegant, customizable theme for Open edX.



### 4.1.2 Gathering Insights and Analytics about OpenedX with Cairn

Cairn is a scalable, real-time analytics plugin for Open edX. Analytics are an essential component of an online learning platform: we need to know whether our courses are effective and which parts need some improvement. We need to know if our students are falling by, and if they do, we need to detect the early warning signs. When our courses are successful, we want to get periodical engagement reports.

We created a tool to help us answer all these questions. Cairn is a Tutor plugin that we install on top of an Open edX platform and that gives us access to a powerful, full-blown analytics stack. Cairn comes with the following features out of the box:

- **Unified data lake of learner events and stateful data:** both learner-triggered events, coming from the Open edX tracking logs, and stateful data, coming from the existing databases, are available for querying in a single unified interface. This means that we can, for instance, query the grades of the students who visited our

platform in the past 24 hours, or collect the email addresses of the students who did not yet complete the latest assignment.

- **Real-time:** new events are visible immediately in our analytics interface. No more waiting for slow batch jobs to complete!
- **Course- and org-based data access rights:** our course staff is granted access only to the data rows that concern them. Cairn makes it easy to create new users with granular access permissions.
- **Working dashboards out of the box:** Cairn comes with a fully functional dashboard that we can start playing with right away.
- **Fully customizable data and dashboards:** our data scientists, business intelligence team, and other tinkerers can freely explore our course data, and create and share their own queries, datasets, and dashboards. All it takes is a little bit of SQL.
- **Scalable:** Cairn scales as much as its backend, which was designed for Internet scale.

Cairn uses the same collect/store/expose paradigm made popular by other frameworks such as the ELK Stack -- except that all the components are different and better suited to Open edX:


- 
- On the server side, tracking logs are collected by Vector, an efficient, cloud-native log collector.
  - Tracking log events are then stored in a Clickhouse table, which is the cornerstone of Cairn. Clickhouse also exposes MySQL data via live and materialized views. This is the magic piece of the puzzle that allows us to join events and MySQL data.
  - The data inside Clickhouse is made visible to the end-users in a Superset frontend.



Figure 24: Cairn overview

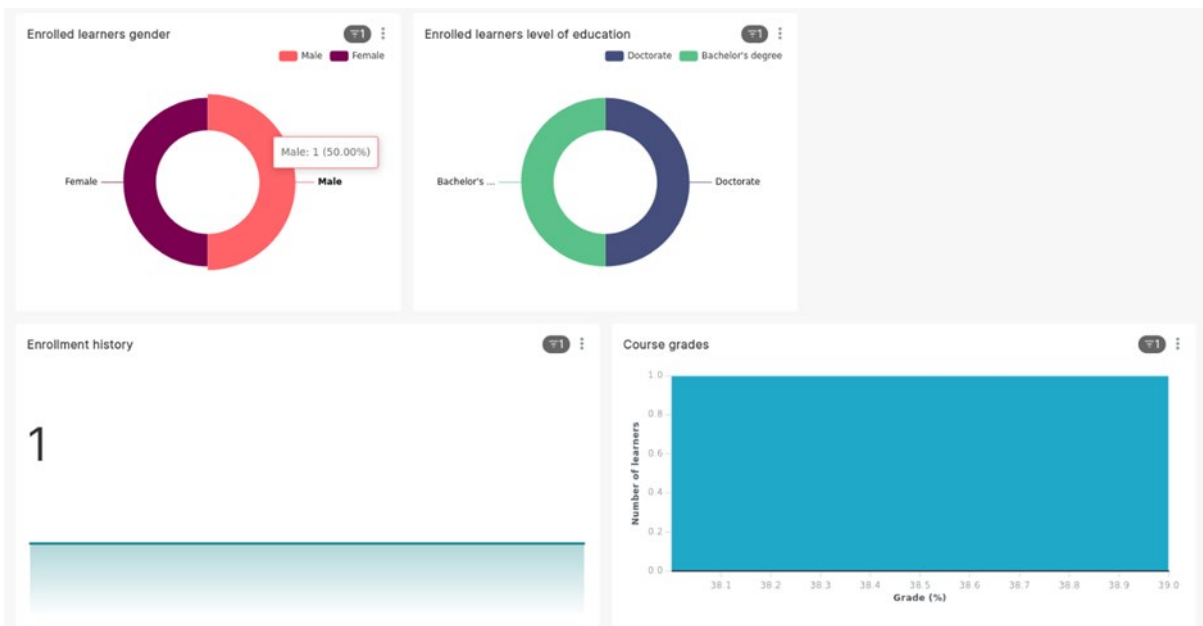


Figure 25: Cairn dashboard

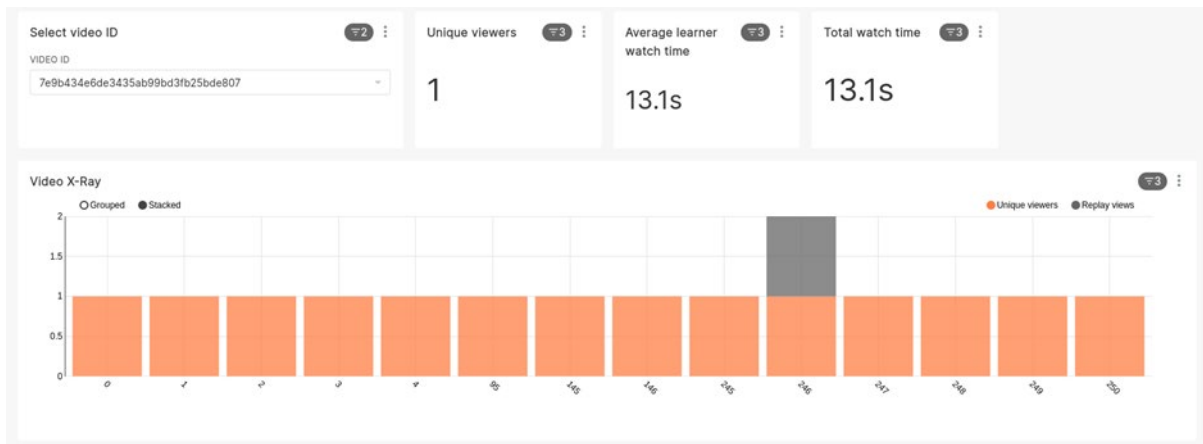


Figure 26: Cairn analyze

Cairn allows us to collect and view just any metric from our Open edX platform, as long as the data is available from the tracking logs or the MySQL database. Out of the box, the default Cairn dashboard comes with visualizations for the following metrics:

- Weekly course engagement:
  - Number of enrolled learners
  - Number of active learners
  - Number of learners who watched a video.
  - Number of learners who tried a problem.
  - Average time spent in the course.
- Course progress:
  - Number of students who completed each unit.
  - Grade distribution histogram
- Course demographics:
  - Gender distribution
  - Level of education distribution
- Video engagement:
  - Number of unique viewers
  - Average watch time
  - Total watch time
  - Second-per-second statistics: Number of unique viewers, Total number of views

## 4.1.3 CodeJail

Tutor plugin that enables execution of untrusted code in secure sandboxes using an external service based on the [codejail](#) library.

The screenshot displays the CodeJail plugin interface. At the top, there are navigation options: 'Live Examples', 'IntegralGrader', and 'Motivating Example'. A 'View Live Version' button and a 'Preview' button are visible. Below the navigation is a toolbar with 'Previous', 'New Unit', and 'Next' buttons. A caution message states: 'Caution: The last published version of this unit is live. By publishing changes you will change the student experience.' The main content area shows the 'Motivating Example' unit with a description and a graph of the function  $y = e^x$ . The graph shows the area under the curve from  $x=0$  to  $x=1$  shaded in blue. The text below the graph asks: 'Write the area of the shaded region as a definite integral.' The right sidebar contains 'Published and Live' information, including the last published date (Apr 23, 2022 at 02:58 UTC by admin), release date (Jan 01, 2017 at 10:00 UTC), and visibility settings. The 'Unit Location' sidebar shows the location ID 'v-integral1' and instructions for linking to the unit.

Figure 27: Codejail plugin with success

The CodeJail service provides a sandbox to run arbitrary code. Security enforcement in the sandbox is done through AppArmor, this means that AppArmor must be installed in the host machine and the [provided profile](#) must be loaded.

The plugin provides an init task that runs a privileged container capable of loading the needed AppArmor profile into our machine. This is only compatible with a docker installation. In Kubernetes we must guarantee that each node of our cluster has AppArmor installed, and the profile loaded, for that reason the one-time initialization task that is used in the init is skipped when running on Kubernetes.

We need to run `tutor local do/dev init --limit codejail` for apparmor profile, if not error in Studio will arise:

Error formatting HTML for problem:  
cannot create LoncapaProblem block-v1:edX+DemoX+Demo\_Course+type@problem+block@integral1: Error while.  
executing script code: Codejail API Service is unavailable. Please try again in a few minutes.

The screenshot shows the IntegralGrader interface for a 'Motivating Example' problem. The error message is displayed in red text: 'cannot create LoncapaProblem block-v1:edX+DemoX+Demo\_Course+type@problem+block@integral1: Error while executing script code: Codejail API Service is unavailable. Please try again in a few minutes.' The interface includes a navigation bar with 'Previous' and 'Next' buttons, a 'View Live Version' button, and a 'Preview' button. A 'Caution' message states: 'The last published version of this unit is live. By publishing changes you will change the student experience.' The 'Published and Live' sidebar shows the last published version on Apr 23, 2022 at 02:58 UTC by admin, and the release date as Jan 01, 2017 at 10:00 UTC. The 'Add New Component' section includes buttons for Discussion, HTML, Open Response, Problem, and Video. The Vietnamese-German University logo is visible at the bottom of the component area.

Figure 28: Codejail failure state

## 4.2 Instructor

The screenshot shows the 'Studio Home' interface. At the top left is a navigation icon. At the top right are 'Help' and 'khoikieu16...' links, and two green buttons: '+ New Course' and '+ New Library'. Below the header is a 'Studio Home' title. Underneath are tabs for 'Courses', 'Archived Courses', and 'Libraries'. The 'Courses' tab is active, displaying a list of five course entries. Each entry includes the course name, a breadcrumb path (e.g., 'edX / DemoX / Demo\_Course'), and two action links: 'Re-run Course' and 'View Live'. The right sidebar features a 'New to EDUNET - Studio?' section with a paragraph of text and a 'Getting Started with EDUNET - Studio' link.

Figure 29: Instructor page

## 4.3 Learner Vietnamese-German University

The screenshot shows the 'Learner' page for a course. At the top left, there is a navigation icon and the breadcrumb 'AASolutions: AAS1 / Test Vocabulary / Discover New'. At the top right are 'Help' and 'khoikieu1608' links. Below the header are tabs for 'Course', 'Progress', 'Dates', 'Discussion', and 'Instructor'. The 'Course' tab is active, showing a breadcrumb 'Course > Test > Subsection > Unit'. Below this is a navigation bar with '< Previous', a document icon, another document icon, and 'Next >'. The main content area is titled 'Unit' and includes a 'Bookmark this page' link. Below that is 'Lesson 7.1 Vocabulary' with '10 points possible (ungraded, results hidden)'. The instruction says 'Fill in the blank with the correct option'. The text reads: 'Running a no-frills airline is never easy as the competition increases and prices for flights keep falling. When flying started in the 1930s, every customer was given VIP \_\_1\_\_ and received personal \_\_2\_\_ from flight attendants. Today, you are lucky if you get a free glass of water with many airlines. Their staff often seem \_\_3\_\_ about the apparent lack of customer care. In fact, their attitude is that if you pay so little for a flight, you shouldn't expect a \_\_4\_\_ service. And if things go wrong, staff can be very \_\_5\_\_. Many customers are extremely \_\_6\_\_ with the way some of these airlines handle problems. Furthermore, the additional costs for \_\_7\_\_ boarding or extra baggage make the ticket price comparable with traditional airlines. As a result, customers can lose \_\_8\_\_ in these companies. Despite the \_\_9\_\_ caused to customers and all the \_\_10\_\_, it is doubtful that cheap airlines will disappear anytime soon.' Below the text is a 'Blank 1' question with a dropdown menu labeled 'Select an option'.

Figure 30: Learner page in a course

## Chapter 5: Limitation and Future work

### 5.1 Limitation

During the development of this project, most of the main function are completed but there are many problems encountered and new requirements should be considered:

- Environment: All of our page run at the local and I am not having much knowledge to make the OpenedX platform be accessed publicly because we have not domain name for the webpages
- Installation: Thanks to the instability of the network where I work, I have encountered some problems with node-js error in install Tutor for OpenedX platform like set the min/max fetch-retry timeout, EACCESS error, ...

### 5.2 Future work

As mentioned in the limitations, there are some requirements that I have not fulfilled, and many problems have not yet been solved. All of this should be considered as future work to help provide the OpenedX platform into more universities and educational institutions.

## Reference

1. Aldo Vladimir Sabando Barreiro," Education 4.0 and its impact on the educational system during the pandemic and post pandemic Covid 19 in Ecuador", *Sinergias educativas, Universidad de Oriente, México* , vol. 7, no. 1, 2022.
2. Marwa Mohamed Zalat Mona Sami Hamed , Sarah Abdelhalim Bolbol, “The experiences, challenges, and acceptance of e-learning as a tool for teaching during the COVID-19 pandemic among university medical staff”, *National Center for Biotechnology Information*, March 26, 2021.
3. Claudiu Coman, Laurentiu Gabriel Tiru, Luiza Mesesan-Schmitz, Carmen Stanciu, Maria Cristina Bularca, Online Teaching and Learning in Higher Education during the Coronavirus Pandemic: Students’ Perspective, Multidisciplinary Digital Publishing Institute, Academic Open Access Publishing, Dec 11, 2020.
4. "Stanford to collaborate with edX to develop a free, open-source online learning platform", *Stanford University*, April 3, 2013.
5. Rosenberg, John S. (2023-03-30), "Axim Online Education Venture Debuts", *Harvard Magazine*, 2023-07-05.
6. <https://ubuntu.com/server/docs>
7. <https://docs.docker.com/get-started/overview/>
8. <https://docs.docker.com/engine/>
9. <https://docs.docker.com/compose/>
10. <https://www.python.org/doc/essays/blurb/>
11. <https://www.geeksforgeeks.org/network-programming-python-http-server/>
12. [https://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework))
13. <https://mindmajix.com/introduction-to-react-js>
14. <https://openedx.org/community/>
15. <https://docs.tutor.overhang.io>
16. <https://github.com/overhangio>