# Internal learning Portal for an organization

# General

-System must have responsive design.

-System does NOT need payments, courses are free.

-System does NOT need chat.

-System should support export/backup…whatever is more suitable.

- System must support delete logs (or something for clear up database beyond certain date).

- friendly slug for resources (slug “course” and “lesson” should support parameters – in ENV file slug for course and lesson should be entered).

-Translation: application has 10/15 terms that should be supported in multiple languages – English + local (I will translate) + few system messages (“not authorized”, “course created successfully” etc.). Other pages can be in English, I will customize it according to organization needs. App will have default language and users can choose language.

(in general, such systems does not need language because courses are created in particular language but it is not appropriate to have local app in English).

Features:

* ~~Multiorganization support – admin can manage multiorganization (every organization can have separate logo, css colors, contents inside, footer). Application should be created to have one organization per domain/subdomain. Example. Org1.domain.com is first and org2.domain.com is second org on same installation (server/app/db), access path is different. If you have different proposal to satisfy such requirement please suggest. Maybe to install each organization on different server. If this is too heavy to accomplish in terms of time-effort, I can survive without it. Please separate pricing for this multiorganization feature if it’s too hard to achieve~~.
* authorization on at least 6 roles (admin, supervisor, creator, manager, user, public). ~~Admin is for all organization system parameters, supervisor is on organization level.~~ Public is unauthorized user-any person from internet.
* System should have ENV parameter for public-unauthorized users (for example public=enabled or disabled). If disabled, front page is login page. If enabled, public users can see homepage, courses page and individual course page.
* Users are authorized trough oauth/2 email (office 365, google, pop3/imap -mail system that company uses) It is important that system support federated authorization with all types of email an companies may use. There is no classic user pass (system may support that but we will disable it for security reasons). Email address will be like ID.
* Users can not register, new users are created by admin/supervisor from inside the system (usually admin will load csv file). User can be disabled/banned. User can be completely deleted.

# Objects/tables

Users

Users have following fields:

* Email
* Organization (for multiorganization support)
* Date created
* Date updated
* Date (manual field)
* Groups (groups where user belong: all, marketing, sales…) please see user\_groups
* Role (admin, supervisor..)
* Disabled-banned (T/F) – disabled user cannot login.
* Sign in count
* Last sign in date
* Last sign in IP
* Social links (Users have MS Teams/LinkedIn contacts - to be displayed next to contact names) or next to course author name)
* Courses count – will be needed later for reporting, not mandatory from my side to be here.
* Enrollment count - will be needed later for reporting,…
* Comment count- will be needed later for reporting,…
* User\_lesson count- will be needed later for reporting,…

Users table should support download and uploaded via CSV files. When uploading, if users email exists, he is updated (belonging group, other profile info), if user does not exist – he is created from CSV file.

Roles

* Public: see only public pages, courses, lessons if allowed.
* User: can enroll and take course.
* Manager: can give assignment to users + what user can do
* Creator: can create and track own courses + what manager can do
* Supervisor: can create users, approve creator course when it’s created and oversee system + what creator does
* Admin can do/see everything, including organization setup.

User groups

* system must support user groups. A user can belong to several groups (ex. All-users, marketing, managers etc.) – Groups are related to courses. Course can be in several groups. If course is related to group ‘all’, users that belong to group ‘all’ can see the course and enroll. If course is in group ‘sales’ users cannot see on portal or enroll to that course except if they (users) are in group ‘sales’.
* User group fields:
  + Group name (string) Example: marketing, all users, etc.
  + Courses belonging to group. For example: course 123, course 345. Course can belong to more than 1 group.
  + **Group** manager: email of person who can change group courses or users who belong to group. Supervisors and admin can change every group.

System behavior is that user cannot access or see the course on “homepage” or on “courses page” if user and course do not have matching the group. If you have better design for such feature, that can be better optimized I am willing to accept your approach.

However if the field “course\_landing\_page\_visible” is true, then user can see the course and course landing page with info how to get access “course access info field” but cannot enroll to the course. (basically user can see what this course is about and ask for permission).

* Please note that course creator and author can be different persons. Course creator is only technical person who created the course in the system, while author is just field in course database (not role) and can be different person than creator.
* Video lessons are tracked with percentage of video played. System must support Vimeo or some other video provider (you can suggest here?). This is basis for calculation of total course completed percentage.

Courses

Course is classic video or text course, like on udemy, with chapters and lessons. Course is created by creator (administrative person who creates the course) and has following fields:

* ID
* Name
* Short description (100 char)
* Description rich text that can contain picture.
* Avatar for the course
* Picture - same as avatar but larger, for main course page ie. “landing course page”
* Course language (eng, rus, srb..)
* Level (beginner, intermediate, advanced)
* Published t/f – published by creator
* Approved t/f – approved by supervisor/admin
* Internal/external mark. (external is scorm content, internal is internally course - like udemy). Please see xapi below.
* Groups (all groups where the course is belonging) please see user\_groups
* Category (IT, Marketing..)
* Tags (accounting, windows, photoshop…). Course have categories and tags so that user can browse certain category/tag. New tags can be created by admin or supervisor.
* Course access info (string): Message that will be displayed in the “card body” of the course if user does not have access to the course contents. Example: “please ask HR manager to get access.” – please see “user group” for the explanation of this field.
* ~~Course landing page visible: bool (true/false) – please see “user group” for the explanation of this field.~~
* Creator (automated field populated when someone creates course). Can be changed only by superadmin or admin. (change creator of the course to another user email). Creator is course owner in technical terms – person who uploaded the videos etc.
* Author ( author person email, linked-in link)
* Attachments (pdf, xls, doc..)
* Q&A (QA threads for user/author comments and questions)
* Average Rating (in stars 1-5) – contains relation to rating and user comment about the course
* Certificate (join to certificate record/table). Course may have certificates that are sent to users email in PDF depending of completion with users first/last name on certificate.
* Courses are searchable by title, course description, lesson name, lesson description. Result is course that contain “search term”.
* Course video length / sum of all lessons video length (read only field)
* System should be capable to import Course contents in xapi format (adobe captivate, storyline etc.). System should store basic progress of xapi courses and regular courses in the same table so that user can see that he completed the course.
* Scenario for xapi course:

1. Creator creates the course
2. Graphical user interface, application

   Description automatically generatedHe uploads zip file with xapi course (like 100 mb itsec4.zip file) please note the file structure of zip file:
3. System receives zip and extract contents in some location.
4. When user click to enroll or open the course, index.html is triggered and new browser window is opened.
5. The course will send records to external database. External database will return record back to our LMS via API.
6. System will store the record (i.e. user [john@company.com](mailto:john@company.com) completed the course) to the database (user\_lessons table) and will send associated certificate to user.
7. System will store api record in separate table.
8. Sample statements are in excel file attached



Chapters

Chapters are placeholder for lessons (lesson grouping), no other content except systematization of lessons.

Lessons

* Id, number
* Video, Video link
* Video length - if lesson is video then video length is mandatory field.
* Avatar (picture displaying at the beginning until video is played) – can be first frame of the video
* Attachment (pdf, xls attachments with lesson)
* Content (rich Text – text field for lesson content – if lesson has content besides video it should be displayed like classic html page)
* Sample lesson (T/F) – if lesson is sample than it’s visible to all users. Sample lessons has “preview” mark, like on udemy.

Percentage passed (user percentage of video completion), progress bar, should be remembered.

Certificates

* Certificates are in jpg format.
* After successful course completion system prints user first, last name, date on certificate jpg and sends in pdf format to user email and maybe another mail. CC option for mail should be present.
* Must contain coordinates to print name and date.
* can be landscape or portrait.

Assignments

* User can receive assignments (course that he/she must enroll). Example – all users must enroll course ABC, or John must enroll course DEF, or user\_group must enroll to course GHI. In case user have an assignment it will be displayed to him when he/she is logged in (explained in pages description)

Enrollments and user\_lessons:

* Enrollments table should remember all enrollments and user\_lessons should store progress for particular lesson. If lesson is in video format, system should remember how much of the video user is viewed and remember percentage. If video lesson is viewed >90%, system should consider that lesson is viewed 100%.
* Course completion percentage is calculated as percentage of “viewed time”/”all lessons time”\*100%.
* When system receives message trough API from xapi, system should record relevant fields in user\_lesson table. Xapi course can be 0% or 100% completed.

Scorm/Xapi

* Xapi course will receive completion record from API which will be developed and accept the record from external database. System should insert that record in User\_lesson, so that particular xapi course can be 100% completed.

# Web application pages

1. Login page – buttons:

- login with your account (office365)

- login with your account (google)

- login with your imap/pop account

2. Homepage:

Page sections:

* Some jumbotron
* Most popular courses (3 suggestions)
* Last updated courses (3 suggestions)
* High rated courses (3 suggestions)
* User comments (3 user course comments)

3. Courses page:

* Below menu is List of category (like row with buttons/pills)
* Advanced search box, you know this better than me.
* List of curses (9) with pagination

4. My courses page (with badge of number of current assignments for the user):

* List of assignment courses that are assigned but not finished
* List of courses that employee enrolled sorted by last accessed date

5. Activities page (only for last 60 days):

* List of all activities in the organization: enrollments, course (create, update, delete), lesson (create , update, delete), comments posted or deleted.
* Supervisor and admin should have access to activities page.
* Activities page should have filter at the header and pagination at the bottom.

6. Enrollments page:

* All course enrollments, comments, lessons visit should be listed on this page.
* Course creator should have access to enrollments on his course.
* Supervisor and admin should have access to enrollments page

7. Single Course page:

* Landing page on left column and lesson on the right (similar to udemy)
* Below video screen there are course contents tab, attachments tab, comments tab.

Header menu:

All users see following:

1. Logo png/svg (homepage link)
2. Courses page
3. My courses
4. Search box
5. Profile/logout
6. Creator (creators, superadmin and admin role can see this menu item)
   * Create new course
   * Course list (courses he created so that he can click and edit course)
   * Enrollments (Creator has access to enrollments of his courses) which represent subset of table with filter and columns: date/time, course name, user/student, user-ID, rating, comment, progress, delete-enrollment-link. Supervisor and admin have access to same menu item, but with all courses.
   * For Xapi course creator can see which user has completed the course (data from API interface).
7. Admin menu (only for supervisor and admin)
   * “User list” – list of users with filters and with link to add user, edit user, Fields in list are id, user email, created date, online status, last sign in, last ip, courses count, enrollment count, comment count, lesson count, user roles, user groups.

Export\_csv\_users, Import\_csv\_users – only admin role can perform this actions (see this buttons), not supervisor.

* + User group (add /edit/delete group)
  + Activities with fields: date time, object (course, lesson,comment,enroll), create/update/delete actions, who performed it user@email.
  + ~~Organization setup (admin only, no supervisor – it can be located elsewhere, for now I put org. setup here)~~
  + Unapproved courses list – courses published but not approved by supervisor/admin with course name and link to course in order to approve it.
  + Category management page (add,edit,delete).
  + Tag management page (add,edit,delete).
  + Database menu (only admin) - Complete Backup and restore database submenu Purpose is to be able to backup and create same database in testing environment on different system to try risky operatons (load csv over existing users etc).
  + Xapi data (only admin). All data received from api (should be stored in separate table) with filters on course and user

1. Profile page – available to all users. Contains basic data of user with list of courses and certificated he enrolled with percentage of completion. (I can show such page sample)