Phase 1

Brief problem statement

To make a recommendation engine for Gradwise which is a learner centric model learning platform. This recommendation engine will suggest the next step that has to be taken by the user depending on the previous performance(like tests taken before). This suggestions include

1) revising a previous topic.

2) giving a test for the current topic.

3) starting to learn the next topic.

The project scope also involves attaching various tags for the questions(which can be of any form of media i.e. text, video, bot exercise) in the practice or quiz sets. This will make the preparation of quiz much easier and simpler, and less restricted. The project focuses mainly on course progress, current skill levels for various topics, learning methodologies modified with different type of courses. And any possible changes in the course structure should be handled.

System Requirements

1)Creation of Database schema

Database schema to store all the details of the courses, exercise data, along with the tagging system is to be implemented in tables with interconnecting tables along wise. We use MYSQL for the creation of database.

2) Project Creation

A sample project is created with API and UI for crud operations on the database. This will allow the system administrators and the trainers to change the course structure at any point of time. We use Node.js, HTML, CSS for this.

3) Creating content skeleton

The entire project content has to be represented, using a simple data schema and sample test data. We use REACT platform to create a sample project.

4) Allowing MCQ and text base content

While designing the exercise tables, there are mainly two types of contents present.- MCQ’s and text based formats. There can be other formats as well, such as bot exercise, theory question, coding question etc.

5) Interlinking content

All the tags and the content must be interlinked with each other. Mainly in the format as follows: Course→ module→sub skill→skill topic. Each topic can have pre-requisites connected to them, which will be further used in tagging.

6) Adding meta tags to the exercises

All the exercises are mapped to tags of course, module, sub skill, and skill topic, level, type of exercise, content type. All these tags are extracted by joining many tables together in the database, and establishing a connections and mappings.

No.	User Case Name	Description	Release
1.	Meta data of courses	To store the learning content in an orderly manner for easy flow of CRUD operations	R0
2.	Exercises data	Designing a place to store all the data of all the practice material, which has to be easily altered.	R0
3.	Exercise content	Able to fit any type of content, including images, videos	R0
4.	Tagging	Meta data of questions to store and map all the possible tags for a particular exercise.	R0
5.	API	Used for access of database and make any possible changes using Node.js	R1
6.	UI or HTML page	Making a accessible website to perform all the CRUD operations on backend.	R1

7.	Populating the data	Populating all the table to see if all the functionalities are working fine.	R1
8.	CRUD operations	Add, Delete, Update, Read operations on the database content	R1

Use case description

Use Case Number:	1
Use Case Name:	Metadata
Overview:	To store the learning content in an orderly manner for easy flow of CRUD operations. This includes, courses, modules, sub skills, skill topics.
Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	This is the first use case that is going to implemented, so there is no pre-condition.
Flow:	This is the first step after start.
Post Condition:	After this step, we must create the database of exercises.

Use Case Number:	2
Use Case Name:	Exercises content
Overview:	Designing a place to store all the data of all the practice material, which has to be easily altered.
Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	precondition is to fill the data of the learning material
Flow:	To fill the data types of the practice material.
Post Condition:	After this step, we must make the tables to store the exercise content.

Use Case Number:	3
Use Case Name:	Exercises data
Overview:	Able to fit any type of content, including images, videos. and making every question connected to its content.

Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	precondition is to fill the data of what is the practice material.
Flow:	To fill the data or content to be shown when the question is seen by the user.
Post Condition:	After this step, we must make the tag all the exercises with their respective tags.

Use Case Number:	4
Use Case Name:	Tagging
Overview:	Meta data of questions to store and map all the possible tags for a particular exercise.
Actors:	Trainers of the learning platform and platform administrator. Useful for trainees and trainers, to organise the questions.
Pre condition:	To fill the data or content to be shown when the question is seen by the user.
Flow:	To map each question to their respective tags in meta data.
Post Condition:	After this step we must make an API which that access the database.
	After this step we must make a UI which that connects to the middle ware using a HTML page.

Use Case Number:	5
Use Case Name:	API
Overview:	After this step we must make an API which that access the database.
Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	To map each question to their respective tags in meta data.
Flow:	Use all the above data, and perform operations on them.
Post Condition:	Populating the table.

Use Case Number:	6
Use Case Name:	UI design

Overview:	Making a accessible website to perform all the CRUD operations on backend.
Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	precondition is to fill the data of the learning material
Flow:	To map each question to their respective tags in meta data.
Post Condition:	Populating the table

Use Case Number:	7
Use Case Name:	Populating the database
Overview:	Populating all the table to see if all the functionalities are working fine.
Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	Completing API and UI design making a feasibility to populate
Flow:	filling the respective data using the UI made.
Post Condition:	Check the working of the complete design.

Use Case Number:	8
Use Case Name:	CRUD operations
Overview:	Add, Delete, Update, Read operations on the database content.
Actors:	Trainers of the learning platform and platform administrator.
Pre condition:	All the tables must be populated.
Flow:	Check all the functionalities using test cases.
Post Condition:	Go to phase 2 implementation.