***EPTL training tool*** Technical Design Document

# 1. Tech Stack

Languages used for deployment:

* Frontend: React
* Backend: PHP

Databases used for this project:

* In general, databases that support SQL should be used. At this time, our project uses MySQL, but you can use alternatives like Postgres, CockroachDB, MariaDB if you would like.

Source control:

* We use GitHub.

Tools, libraries, IDEs:

* Tools: We use docker to make local deployments faster, and more convenient.
* IDEs: We wrote our backend using Sublime text editor, and the front end using WebStorm.
* Libraries/frameworks
  + Backend: Yii,
  + Frontend: React, Material-UI, Material-Table
  + API: swagger

Dependencies, and versioning:

* We are assuming the latest versions at the time of development. Most of the libraries we use ensure backward compatibility, so specific versioning is not required.

Hardware requirements:

* There are no specific RAM/Disk space requirements at the time of writing this page. As we work in deciding whether the artifacts are going to be self-hosted vs remotely hosted, we will modify this section to reflect the changes

# 2. Accounts and Infrastructure

We only need a single account for the deployment of the production solution of our project on the Stark Server.

2.1 Development

Testing is done on the local setup in Docker containers, or directly on the hardware, we do not require a separate development environment on the CSE servers. The local setup Administrator account will have the following credentials(START = DEV, LANISTER = PROD):

* Username: admin
* Password: admin

## 2.2 Production

We only need a single account for the deployment of the production solution of our project on the Stark Server.

# Data Sources, Models, Timing

## 1.1 Data Sources

Our data sources would-be users who use the application. Primarily:

* Contributors, who are graded by Evaluators, will upload artifacts (videos, papers, code snippets, etc).
* Evaluators who will grade the artifacts
* Managers, who will be responsible for crafting Criterias, and managing users

Data will be maintained by managers of evaluation sessions. Once the evaluation session is over, it will be archived, and then deleted. Deleted evaluation sessions will result in (CASCADE)ing of the artifacts that belong to the evaluation session

## 1.2 Data Models and Structure

Table name:artifact\_info

Description: Contain the information of the artifact like video, pdf, code.

* artifact\_info\_id:

Description: The ID of the artifact.

* title:

Description: The title or name of the artifact.

* description:

Description: The description of the artifact.

* uploaded\_by:

Description: The name of the user who uploaded this artifact.

* address:

Description: The address of the video or pdf or code where it is stored on the server.

Example: “upload/video/\*\*\*\*.mp4”

* session\_id:

Description: The **foreign key** of the table grading\_session.

* evaluation\_id:

Description: The **foreign key** of the table evaluation.

* type:

Description: The type of artifact.

Example: PDF, Video, code.

Table name: comment

Description: Contain the comment for each artifact. Ultimately, after the evaluation is filled up, it is converted into a comment, with a reference back to the evaluation. In case the evaluation is deleted, the reference to evaluation would be set to null, essentially making a generic comment.

* comment\_id :

Description: The ID of the user.

* text:

Description: The name of user and username to login.

* color:

Description: The color corresponded to the comments.

* artifact\_id:

Description: The **foreign key** of the table artifact\_info.

* evaluation\_id:

Description: The **foreign key** of the table evaluation.

* user\_id:

Description: The **foreign key** of the table user.

Table name: user

Description: Contain the information of the user for their role and login information.

* auth\_user\_id:

Description: The ID of the user.

* name:

Description: The name of user and username to login.

* password:

Description: The password for the user to login.

* email:

Description: The email of each user.

* global\_role:

Description: The role of the user for the whole system. This is different from the role that is specific to evaluation sessions.

Example: Contributor, admin, manager.

Table name:grading\_session

Description: Contain all the grading\_session that the manager creates.

* grading\_session \_id:

Description: The IDof each evaluation session.

* name:

Description: The name of each evaluation session.

Table name:session\_role

Description: The role of the user in a certain session.

* auth\_user\_id:

Description: The **foregin key** of the table user.

* grading\_session\_role:

Description: The role of user for each kind of evaluation session.

Example: Evaluator, Owner, Contributor, Manager

* grading\_session\_id:

Description: The **foregin key** of the table grading\_session.

Table name: evaluation

Description: Evaluation represents a collection of single criteria.

* evaluation\_id:

Description: The id of the evaluation.

* name:

Description: The name of each evaluation.

Table name: criteria

Description: Contain the information on the criteria.

* criteria\_id:

Description: The id of the criteria.

* description:

Description: The description of the criteria.

* evaluation\_id:

Description: The **foreign key** of the evaluation table.

* rating\_set\_id:

Description: The **foreign key** of the table rating\_set.

* attribute:

Description: The attribute of the criteria.

Table name: rating\_option

Description: Contain the rating option of the rating for certain artifacts like video, pdf, code.

Columns:

* value:

Description: This is a value of the rating

Example: A, B, C, 1, 2, 3, 5, Good, Bad, Lacking Explanation.

* color:

Description: The color corresponded to the value.

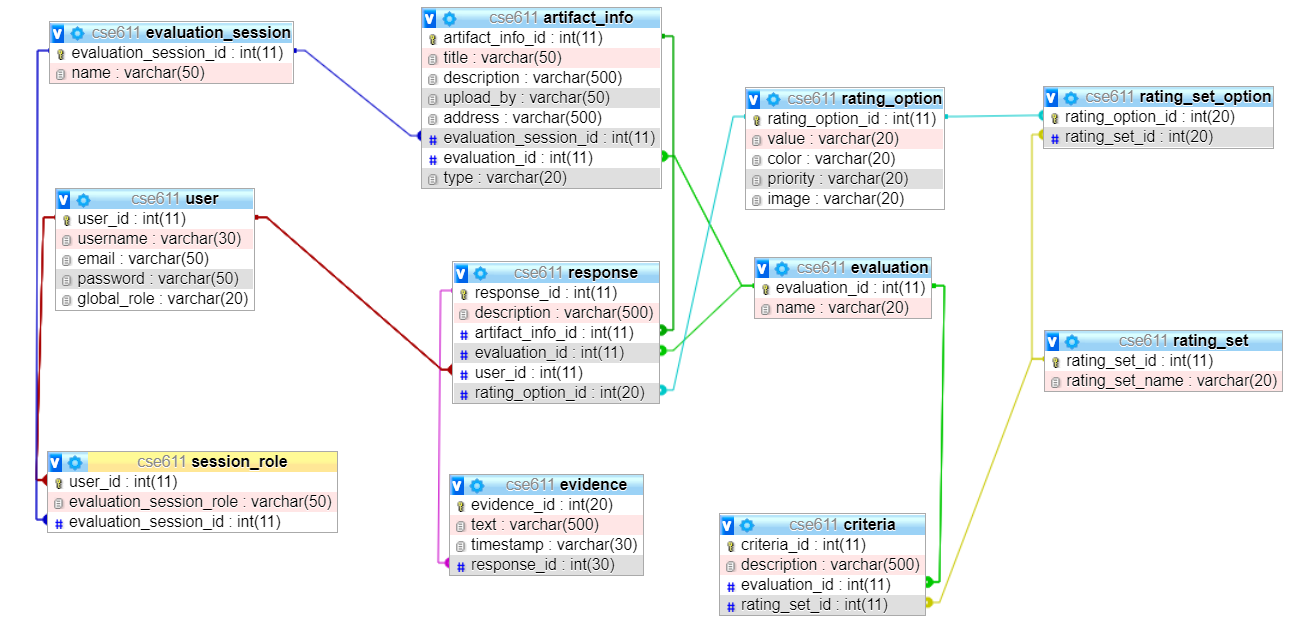
Example: red is corresponded to A, blue is corresponding to B.

* priority: This is the level of the value
* rating\_set\_id:

Description: The **foreign key** of the table rating\_set.

Table name:rating\_set

Description: Contain different types of rating methods.



## 1.3 Timing

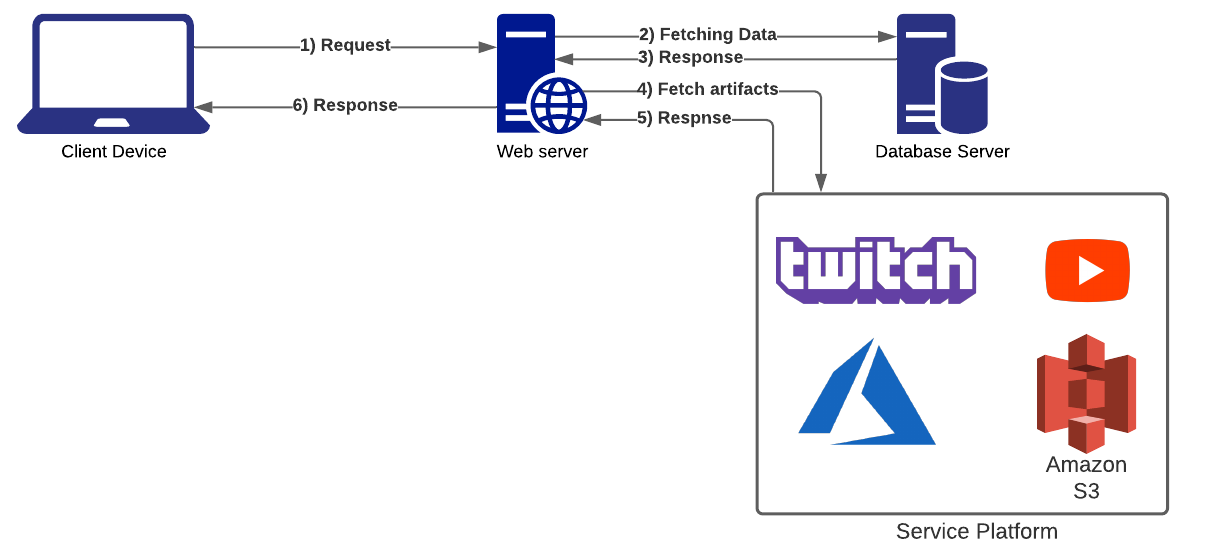
All artifacts belong to a session group. Hence when the session group is removed, all of the artifacts attached to the session group should also be removed. We plan to allow evaluation sessions to exist for a maximum of 6 months, after which they will be archived. A evaluation session can be archived for a time specified in the settings of the application(No archiving, 6 months, Forever, etc). This setting can be managed by Administrators, who will have a better context of the situation (Storage limitations, billing, etc)

# System Architecture Diagram

You can access the front end diagram via the following link: [Figma Diagram](https://www.figma.com/file/tJjzSgwZXKJqt7Sw9XUT4y/Online-Teacher-Training?node-id=0%3A1)

You can access the API representation of the project using the following link [Swagger](https://app.swaggerhub.com/apis/L1ghtman2k/EPTL-Training-Tool/1.0.0), or by using a file at the root directory of the project called api.json (THIS WILL BE REPLACED AFTER THE MEETING)

Following is the diagram of the system:



# Deployment Methodology

Deployment methods:

* There are many different ways we could deploy the application. Depending on the time constraints, this could range from a simple installer that assumes the presence of a LAMP stack, and just installs the application, to a full-fledged deployment using tools like Ansible that will take care of deploying the environment(LAMP stack) and the application.

Turnover:

* Turnover to sponsor should be a relatively simple procedure. We would have to pass credentials to every system that is a part of the application. The production environment will have no data on it, so it will have to be seeded.