Project Overview:

The final project for this training program is a group project that will allow participants to develop a web application as a way of demonstrating their knowledge of enterprise software development, as well as their skills in project management and database development.

The team will work together to build the web application and to create the project deliverables.

Learning Objectives

By the end of this assignment, learners will be able to:

* Work in a team to solve a problem.
* Design and develop a full stack web application.
* Present their application to an audience.

Group Expectations

Each member of the group is expected to contribute actively during all stages of development. Activities should include:

* Live, online discussions using Slack.
* File sharing using Github (especially for code files).
* Collaborating on the submitted files.

Project Submissions

* Use team repo on github.

Group Presentation

Each team will present their final project to an audience via Zoom.

* The presentation will include a team introduction, the topic goal, the background, the need, the development, the management of the project, and the "lessons learned."
* The presentation will be approx. 15 minutes long.
  + The slideshow presentation can be developed using Microsoft PowerPoint, but the use of PowerPoint is not required.
  + Each team member will contribute to a portion of the presentation/demo.

Project Details

Throughout this training, participants will use what they have learned to build a full-stack web application.

For the purpose of this project, each group will develop their own application on an appropriate subject of their choice.

Web Application Requirements

The web application must include the following:

* (optional) A data source for the project
  + Examples include but are not limited to:
    - Amazon Review Dataset
    - [data.gov (Links to an external site.)](https://data.gov/)
* A database
  + Data generated and consumed by the application must be stored in a database.
  + Application must be able to perform all CRUD operations.
* (optional) A dashboard page with visualizations
  + Data generated and consumed by the application may be easier to interpret if visualized. Add graphs, charts, or maps to your application.

Backend Requirements

The backend is suggested to be written in Java/Spring.

* Use an [API-First Development Approach (Links to an external site.)](https://swagger.io/resources/articles/adopting-an-api-first-approach/) for the backend.

Frontend Requirements

Suggested technologies include (but not limited to) HTML, CSS, JavaScript, React, Thymeleaf.

* The interface must be clean, well-laid out, and user-friendly.

Examples from past projects:

App for an airline to track aircraft.

Shopping list app that also creates menus/meals.

App to sell artistic works.

App to track characters for RPG game.

App to track stock trades.

App to track car accidents.