SLOTracker 1.2 Project Design

Version 1.2

Designed By:

The Fall Class of 22022
CSCI 36

Summary

This design is based on beginning requirements and the first sprint of development. This document will detail what the project is and how it works. This document gives a detail description of the files that make up the SLO Tracker project and the requirements the project is based on.

Student Learning Outcomes:

The SLO Tracker is a class project, this design progresses the following Student learning outcomes:

1.Work with a team to develop and deliver a software system that satisfies a specific set of requirements.

2. Analyze and determine requirements for a software system to solve a specific problem.

3. Produce a detailed design for a software system that meets a specific set of requirements.

4. Write software documentation that is clear, concise, and consistent with industry standards.

Table of Contents

Introduction

The SLO Tracker Project is a two-part website and web scraper program developed by students as a class project. The web scrapper searches through the Butte.edu website and collects information. The website will display that information for the convenience of any end users. The SLO Tracker design is made up of multiple files of code that uses a combination of JavaScript, HTML, CSS and EJS. The following pages will detail the design of the SLO Tracker project with explanations charts and diagrams followed by the requirements.

The files

The files that make the SLO Tracker project are can be summarized as follows:

Database - The files in the database folder contains two files database.js and sequlize.js tighter these files set up and fill the database.

Documentation - The Documentation folder contains file contains read me files, (SRS)Requirements, flow charts and other document for the project.

Node Modules -This folder contains all the packages listed as dependencies in the package files.

packages - The project contains package files that are important parts to the foundation of the project. The files package are lock.json, clos.json package.json the contain the libraries/dependencies that the project will use.

Public- The public folder has the files and code to create and style the discussion form page of the website along with style and image files.

Routes -Routes is folder in the project that contain four files departments.js prorams.js, form.js and index.js. These files contain the code that will handle requests from the browser/client to deliver and display the webpages.

Models – The models folder contains multiple files with code that will organize and arrange the data into models. When data is pulled from the Butte college website it is stored in the database using these models.

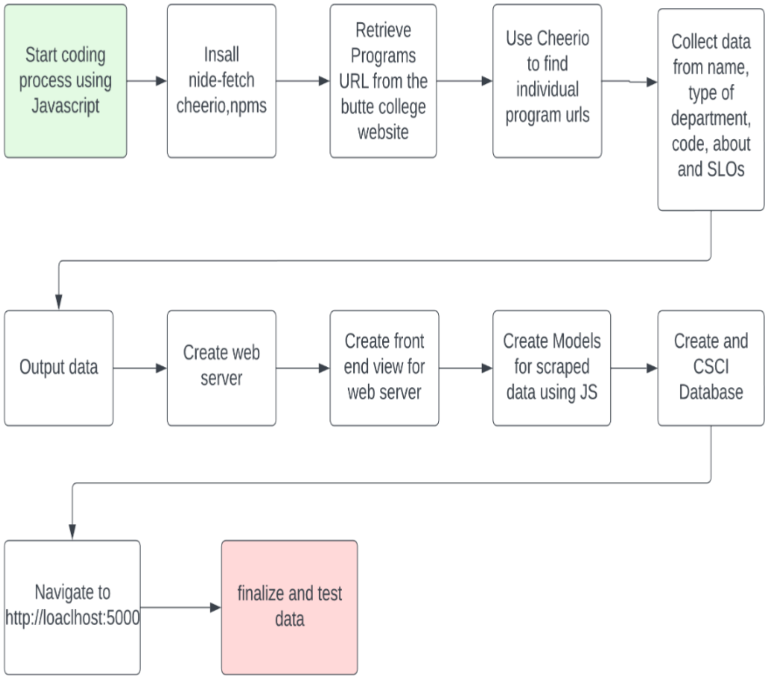
Utilities – helper files for the fetching and parsing functions used by the scrapper and other supporting files.

Views - The views folder contains multiple files of code that make up the design/view of the website.

Scraper – The scraper.js file is the heart of the program because it pulls the information that will flow to other parts of the program. The scraper will be discussed in detail in the next section.

The scrapper

The project contains a file called scraper.ejs. The scrapper file is the contains the code that makes the scrapper work. The scrapper pulls the information that all the other parts of the website will use.


Start coding process using Javascript
Insall nide-fetch cheerio,npms
Retrieve Programs URL from the butte college website
Use Cheerio to find individual program urls
Collect data from name, type of department, code, about and SLOs
Output data
Create web server
Create front end view for web server
Create Models for scraped data using JS
Create and CSCI Database
Navigate to http://loaclhost:5000
finalize and test data

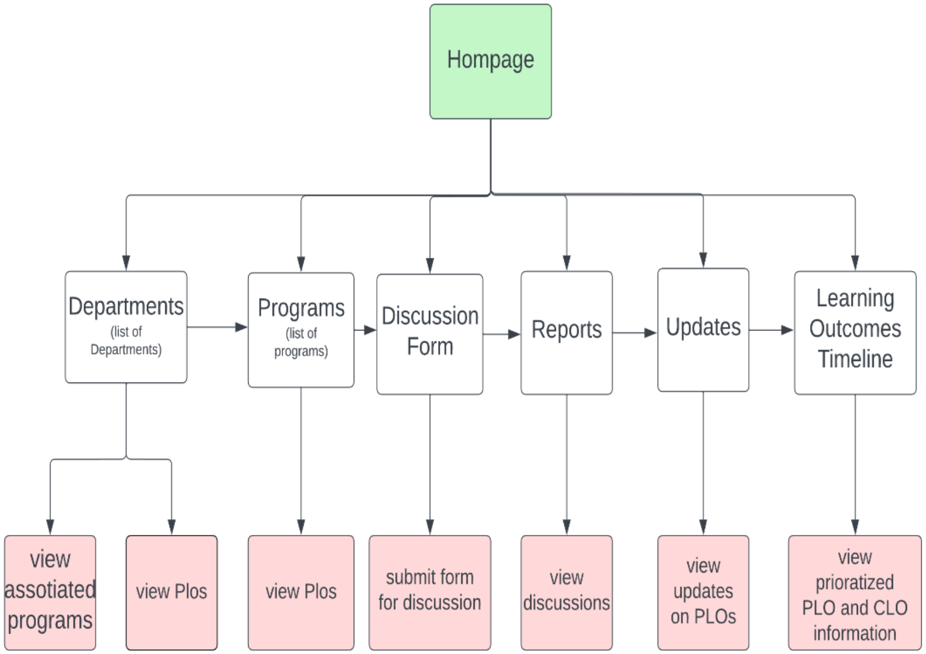
The Scraper flow Chart

To the right is a chart of how the scraper program flows. The program begins with JavaScript installing the libraries/dependencies listed in the package files and contained in the Node module files. The scraper will use the installed fetch and cheerio libraries to fetch the butte.edu web page. The scraper will then search the page and collect program names, departments, types, codes, and links for each program. Next the scraper will then use the programs links to do a second fetch and collect data on the for the SLOs. The program then flows into the sequence of steps that will format and output the data. The Server file uses the JavaScript library express to create the server and handle the routes. The routes are the pages the website will flow into. The models files then create the arranges and organize the information to be displayed on the website. There are error catching and check statements thought the code that will finalize the data.

The Website

Once the data is gathered collected and organized by the scrapper and framed into models, the website will display the data using a combination of web design languages.

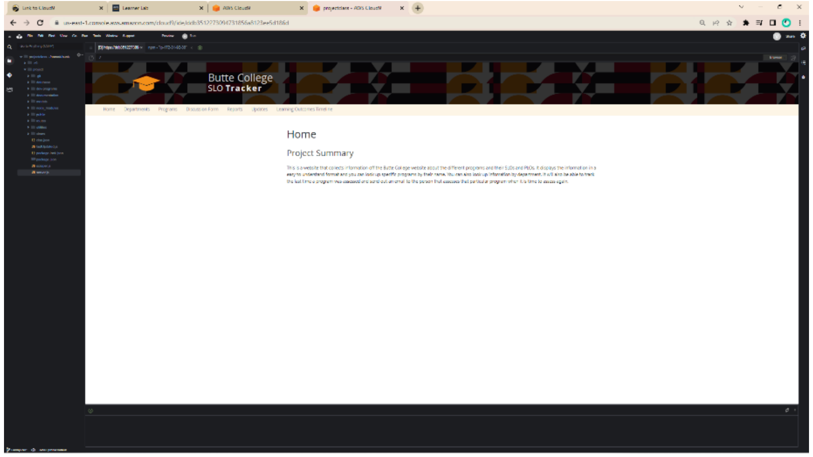
The Website Flow Chart



This chart shows how the website will flow. The website starts with the home page. The home page is that start of the website and gives a brief description of the website. from which users can access the other pages via the navigation bar. The department page will list each department with the name being a clickable link. The link will connect the user to a page that displays a list of programs associated the clicked/chosen department. From this page the user can click a program to view PLOs the programs page is a list of programs offered at Butte college arranged in alphabetical order. On this page the user can click a program to view information about it and PLOs. The Discussion page contains a form that users can fill in to add to the discussion. The reports and updates page will display information on PLOs that have recently been assessed. Lastly the learning outcomes page will display PLOs and CLOs based on their level of priority.

The Website GUI

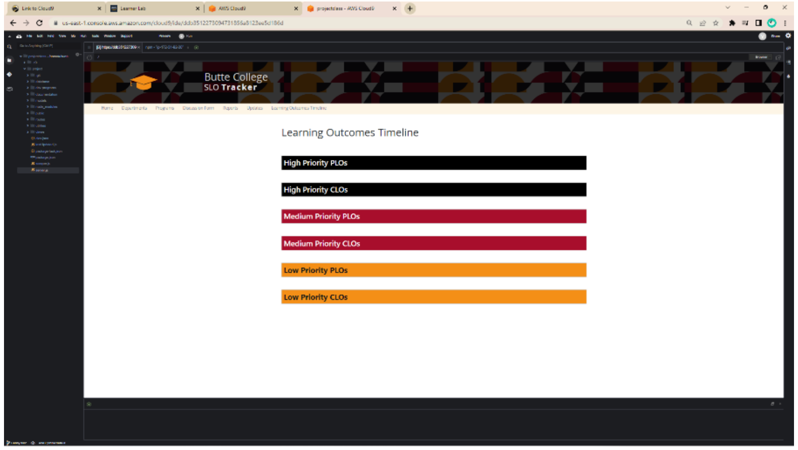
Here a sample of what the website will look like at completion.



Home

Project Summary

This is a website that collects information off the Butte College website about the different programs and their SLOs and PLOs. It displays the information in a easy to understand format and you can look up specific programs by their name. You can also look up information by department. It will also be able to track the last time a program was assessed and send out an email to the person that assesses that particular program when it is time to assess again.



Learning Outcomes Timeline

High Priority PLOs

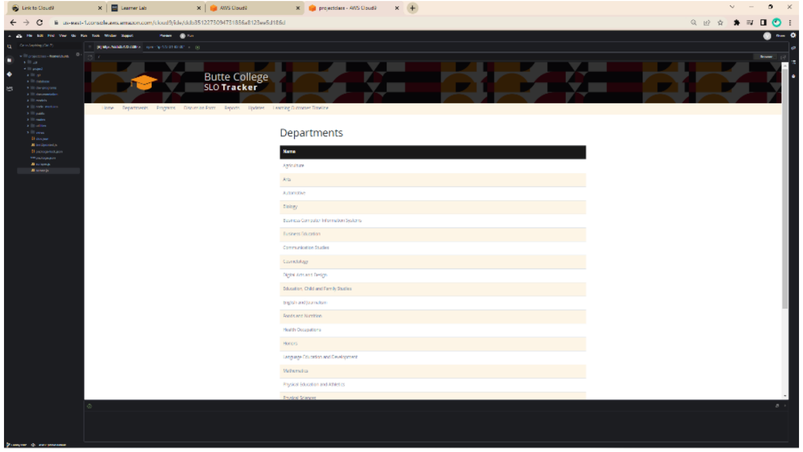
High Priority CLOs

Medium Priority PLOs

Medium Priority CLOs

Low Priority PLOs

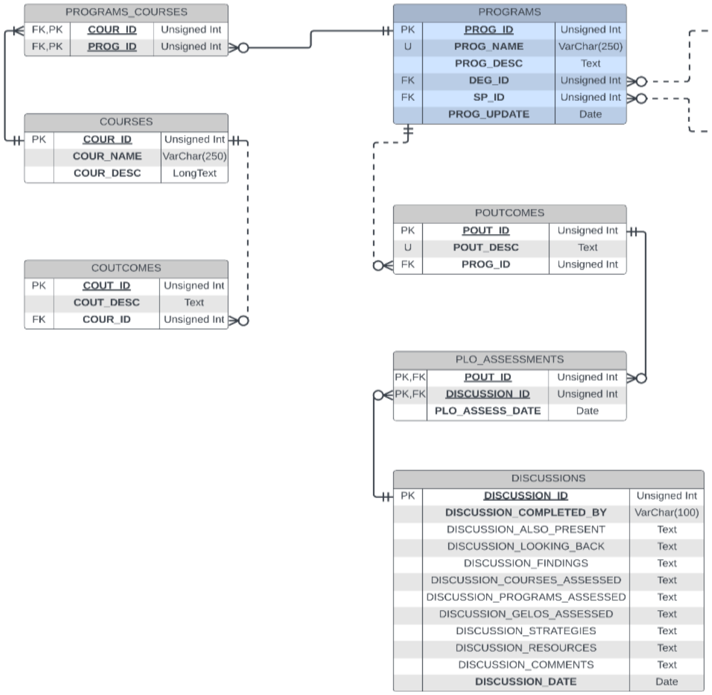
Low Priority CLOs



Departments

Name
Agriculture
Arts
Automotive
Biology
Business Computer Information Systems
Business Education
Communication Studies
Cosmetology
Digital Arts and Design
Education, Child and Family Studies
English and Journalism
Foods and Nutrition
Health Occupations
Honors
Language Education and Development
Mathematics
Physical Education and Athletics
Physical Sciences

The Database



PROGRAMS_COURSES

FK,PK	COUR_ID	Unsigned Int
FK,PK	PROG_ID	Unsigned Int

PROGRAMS

PK	PROG_ID	Unsigned Int
U	PROG_NAME	VarChar(250)
	PROG_DESC	Text
FK	DEG_ID	Unsigned Int
FK	SP_ID	Unsigned Int
	PROG_UPDATE	Date

DEGREES

PK	DEG_ID	Unsigned Int
U	DEG_TYPE	VarChar(25)

COURSES

PK	COUR_ID	Unsigned Int
	COUR_NAME	VarChar(250)
	COUR_DESC	LongText

SUPER_PROGRAMS

PK	SP_ID	Unsigned Int
U	SP_NAME	VarChar(250)
FK	DEP_ID	Unsigned Int

POUTCOMES

PK	POUT_ID	Unsigned Int
U	POUT_DESC	Text
FK	PROG_ID	Unsigned Int

COUTCOMES

PK	COUT_ID	Unsigned Int
	COUT_DESC	Text
FK	COUR_ID	Unsigned Int

DEPARTMENTS

PK	DEP_ID	Unsigned Int
U	DEP_NAME	VarChar(250)
	DEP_CHAIR	VarChar(250)

PLO_ASSESSMENTS

PK,FK	POUT_ID	Unsigned Int
PK,FK	DISCUSSION_ID	Unsigned Int
	PLO_ASSESS_DATE	Date

DISCUSSIONS

PK	DISCUSSION_ID	Unsigned Int
	DISCUSSION_COMPLETED_BY	VarChar(100)
	DISCUSSION_ALSO_PRESENT	Text
	DISCUSSION_LOOKING_BACK	Text
	DISCUSSION_FINDINGS	Text
	DISCUSSION_COURSES_ASSESSED	Text
	DISCUSSION_PROGRAMS_ASSESSED	Text
	DISCUSSION_GELOS_ASSESSED	Text
	DISCUSSION_STRATEGIES	Text
	DISCUSSION_RESOURCES	Text
	DISCUSSION_COMMENTS	Text
	DISCUSSION_DATE	Date

Sequelize is a library listed in the package files that simplifies the process of creating a database. The sequelize.js file contains code that imports the models and uses sequelize to set up a database and format it according to the models. This is an ERD of the how the database will hold the scraped information. The programs connected to the degrees and super programs table. The degrees are the achievements you will receive at the completion of the program. Super programs are the larger programs each degree is a part of. The super programs connect the programs table to the school's departments table. The programs table is connected to the Outcomes table which are the PLOs. The PLOS are linked to the PLO Assessment table and the discussion form. The programs are associated with program courses which connected the to the outcomes table which holds the SLOs This database is designed to for information to flow in and out of the SLO Scrapper project.

THE REQUIREMENTS

Web SLO Tracker 1.2

Software Requirements Specification

1.0 Introduction

Butte College has defined expectations for each course offered called Program Learning Outcomes (PLOs)and Student Learning Outcomes (SLOs). Details about these outcomes are displayed on the Butte College website. SLOScrapper1.0/1.1 were created to collect PLOs from the butte.edu websites and store them in a database because no such program existed. SLOScrapper1.0 was going to be a downloadable program but was discarded and replaced by SLOSCrapper1.1 which is designed to be a website. Since the creation of SLOscrapper1.1 Butte College has made alterations to its website and the information it contains which calls for an updated SLOScrapper. Therefore, this project will be the newest version of the SLOScrapper, version 1.2.

1.1 Purpose

SLOscrapper 1.2 is designed to be a web application that provides a one stop source for a collection of data gathered from the Butte College website. SLOScrapper1.2 is also a class project to meet the following Student Learning Outcomes:

- Work with a team to develop and deliver a software system that satisfies a specific set of requirements.

- Analyze and determine requirements for a software system to solve a specific problem.
- Produce a detailed design for a software system that meets a specific set of requirements.
- Measure and track progress of a software development project.
- Use a version control tool to manage a software project with multiple team members and multiple revisions.
- Write software documentation that is clear, concise, and consistent with industry standards.

1.2 Scope

The fall class of 2022 will design a web application project. The project will be developed using SLOScrapper 1.1 as the foundation. The project will allow users to view information about departments and programs at butte college. Users will be able to find out when a PLO was updated, needs to be updated and will be updated. We also want to allow forms to be submitted by faculty to when a PLO was updated, needs to be updated, or will be updated.

1.3 Definitions, Acronyms, and Abbreviations.

PLO – Program Learning Outcomes
SLO – Student Learning Outcome
Version 1.1 – SLOScrapper1.1 (The previous program) Version 1.2 – SLOScrapper1.2(The program currently being developed)

1.4 Document Conventions

This document has no unusual conventions.

1.5 Intended Audience & Reading Suggestions

This document is written and intended for product owner April Browne, each team member of the fall 2022 Class.

1.6 References

This Project references the overall project document: https://docs.google.com/document/d/1Hp_K6G0nsl3MKC0-4r8pBC7QjkZ7fIy9uiX1XjWn-m4/edit?usp=sharing and our group discord: https://discord.gg/8wuSaGPTYg

1.7 Overview

Section 2 will give on overview of the product and section 3 will go into specific details.

2.0 The Overall description

2.1 Product Perspective

SLOScrapper1.2 is a web application whose primary objective is to display specific information that is up to date and accurate for the convenience of end-users. SLOScrapper1.2 uses a web scraper program to collect information and store the information to be latter retrieved for viewing. SLOScrapper1.2 also uses a combination of web design languages to create and style the web application. SLOScrapper1.2 can be compared with any informational website.

2.1.1 User Interfaces

The user will interact with the web page via navigation bar and clickable links.

2.1.2 Hardware Interfaces

SLOScrapper1.2 will interact with hardware such as computer screens, keyboards, and mouses for browser-based scrolling.

2.1.3 Software Interfaces

We will be using these libraries:

1. Cheerio

a. To scrape the butte college pages

2. Node.js

. To have an actively running server

3. Ejs

. To create modular web pages

4. Sequelize

. To take our data and put into our database

We will use MySQL as our DBMS. We will also use a combination of JavaScript html and CSS.

2.1.4 Communications Interfaces

Version 1.2 requires an internet connection and relies on Node.js and HTTP for our rest API calls.

2.1.5 Memory Constraints

Version1.2 is web application that will require persistent memory.

2.2 Site Adaptation Requirements

Version 1.2 should be responsive to adapt to desktops, laptops, tablets, mobile and other similar devices. There are currently no other known adaptations needed to run version1.2.

2.3 Product Features

Version 1.2 is a class project that effectively does the following:

Meets Student Learning Objectives - The project is functional, and work is tracked and proven at the end of the semester. The project will use NodeJS and scraping libraries to retrieve data from butte.edu.

Implements new functionality - The project tracks department discussions around PLOs the project tracks when assessments of PLOs are planned the project tracks if they've been completed in that semester The project can generate reports of PLOs that are to be planned this semester

Database Implementation - The project implements a database to store scraped objects in. The project should also use the database to track the state of each PLO and SLO.

Implement CI/CD- Software can be set up for continuous integration using GitLab. The project can also be deployed to a server, but a significant amount of additional work is required to accomplish this.

2.4 User Characteristics

Instructors: This user class will most likely use this software infrequently, 2 to 5 times a year to check PLOs. They may use it to ensure that their classes are following the requirements. They may use it to ensure that their classes are following their program requirements.

Administrator: This user class will use the software more frequently.
They will use it when evaluating classes, programs, and institution. They will be able to use the software to ensure that all PLOs have been updated properly. They will be able to use it to see other faculties form submissions regarding updates.

2.5 Constraints

SLOScrapper version 1.2 is not a fully functional website until hosting is provided. Like 1.1, Version 1.2 is not designed to be added to the butte.edu website but designed as an external tool. Version 1.2 will be designed as a website using JSNode, JavaScript, HTML, CSS and any other language deemed necessary by the fall 2022 Class. We do not have any CSS set up for mobile.

2.6 Assumptions and Dependencies

It is assumed that SLOScraper1.2 will one day become the foundation of SLOScraper1.3 therefore, Code is expected to be commented and made clear for other team members as well as future team members. Version 1.2 is completely dependent on The Butte College website http://butte.edu/academicprograms/.

3.0 Specific Requirements

Specific requirements:

1. Version 1.2 shall be a bug free working web application ready for the steps of web hosting.
2. Version 1.2 shall scrape PLO data from the butt.edu website and store it in a relational database.
3. Version 1.2 shall scrape SLO data from PDF's associated with the Butte College website.
4. Version 1.2 shall Update PLOs
5. Version 1.2 shall Update when specific PLOs will be evaluated
6. Version 1.2 shall display which PLOs will be evaluated with the semester and year.
7. Version 1.2 shall display which PLOs need to be evaluated.
8. Version 1.2 shall have a recreated form for tracking PLO discussions
9. Version 1.2 shall Report out PLO discussion information.
10. Version 1.2 should have a new revised more appealing website.
11. Version 1.2 should have a revised what to do section on the discussion page.

12. Version 1.2 should have handicap accessible features

13. Other non-functional requirements.

3.1 External Interfaces

SLOScraper1.2 is designed as a web application capable of running in any browser including but not limited to Google Chrome, Microsoft Bing, Safari Apple, and Mozilla Firefox etc...

3.2 Performance Requirements

The application will perform according to standard NodeJS. The application must be able to fetch the data from the database. The application must also not contain logic errors or other issues that might impact performance.

3.3 Logical Database Requirements

This section will be revised as needed.

3.4 Software System Attributes

This section will be revised as needed.

3.4.1 Security

This section will be revised as needed. Currently, we do not contain any personal information that would be harmful if leaked. All our information are things that could be made public. The worst thing is if the discussion posts were public, and the one who submitted the form didn’t want that.

3.5.2 Maintainability

Version 1.2 will require the PLOs and SLOs to be updated. If the Butte College website changes, then version 1.2 will need to be updated again.

3.5.3 Availability

This section will be revised as needed.

3.5.4 Security

We currently do not have much security, so our product is very vulnerable.

4.0 Other Requirements

This section will be revised as needed.