Course: IT202-010-S2025
Assignment: IT202 Milestone 2
Student: Justin C. (jjc88)
Status: Submitted | Worksheet Progress: 99%
Potential Grade: 9.89/10.00 (98.90%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: https://learn.ethereallab.app/assignment/v3/IT202-010-S2025/it202-milestone-2/grading/jjc88

Instructions

1. Refer to Milestone2 of this doc: https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88EWVwfo/view
2. Ensure you read all instructions and objectives before starting.
3. Ensure you've gone through each lesson related to this Milestone
4. Switch to the Milestone2 branch
    1. `git checkout Milestone2` (ensure proper starting branch)
    2. `git pull origin Milestone2` (ensure history is up to date)
5. Fill out the below worksheet
    - Ensure there's a comment with your UCID, date, and brief summary of the snippet in each screenshot
    - Ensure proper styling is applied to each page
    - Ensure there are no visible technical errors; only user-friendly messages are allowed
6. Once finished, click "Submit and Export"
7. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github
    1. `git add .`
    2. `git commit -m "adding PDF"`
    3. `git push origin Milestone2`
    4. On Github merge the pull request from `Milestone2` to `dev`
    5. On Github create a pull request from `dev` to `prod` and immediately merge. (This will trigger the `prod` deploy to make the heroku prod links work)
8. Upload the same PDF to Canvas
9. Sync Local
10. `git checkout dev`
11. `git pull origin dev`

100%

Section #1: (2 pts.) Data

100%

Task #1 (0.50 pts.) - API Data Table

Combo Task:

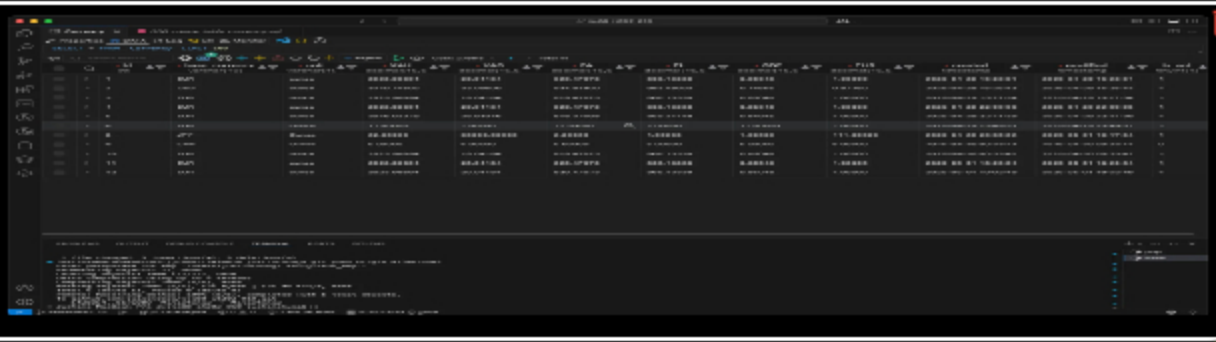
Weight: *25%*
Objective: *API Data Table*

Image Prompt

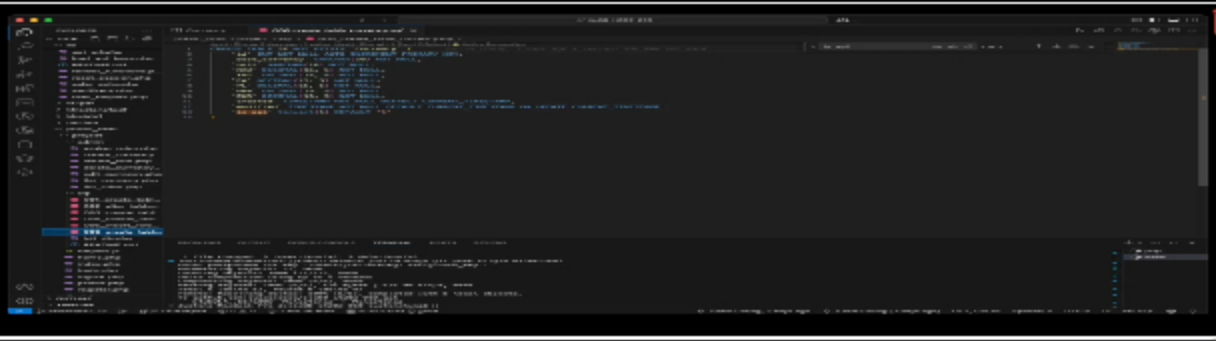
Weight: *34%*

Details:

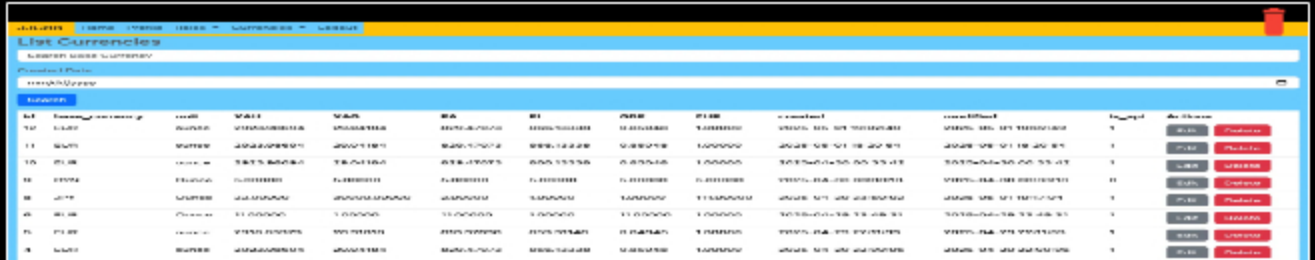
- Show the table(s) structure of the table made to hold your API
- There should be at least 3 properties/fields beyond `id`, `created`, `modified`, and respective `is_api` or `api_id` columns
- Logical types and constraints should be set
- Show at least a few records from each API table (samples should include both API fetched data and custom entries noted by an `api_id` or `is_api column)



db table



sql table file



table

Saved: 5/1/2025 7:33:07 PM

Url Prompt

Weight: *33%*

Details:

- Add a direct link to your sql file from Github (should end in `.sql`)

URL #1
https://github.com/juicejoose/jjc88-it202-001/Milestone2/public_html/project/sql/006_create_table_currency.sql

URL
https://github.com/juicejoose/jjc88

Saved: 5/1/2025 7:33:07 PM

Text Prompt

Weight: *33%*

Details:

- Briefly note what each field/property represents

Your Response:

Id represents the entity number base_currency is the compared currency that the user inputs unit is how the metals will be measured XAU, XAG, PA, PI, GBP, EUR will represent the comparable Created and modified are dates of when created and modified is_api tracks if the data is api or user made

Saved: 5/1/2025 7:33:07 PM

100%

Task #2 (0.50 pts.) - Data Transformation

Combo Task:

Weight: *25%*
Objective: *Data Transformation*

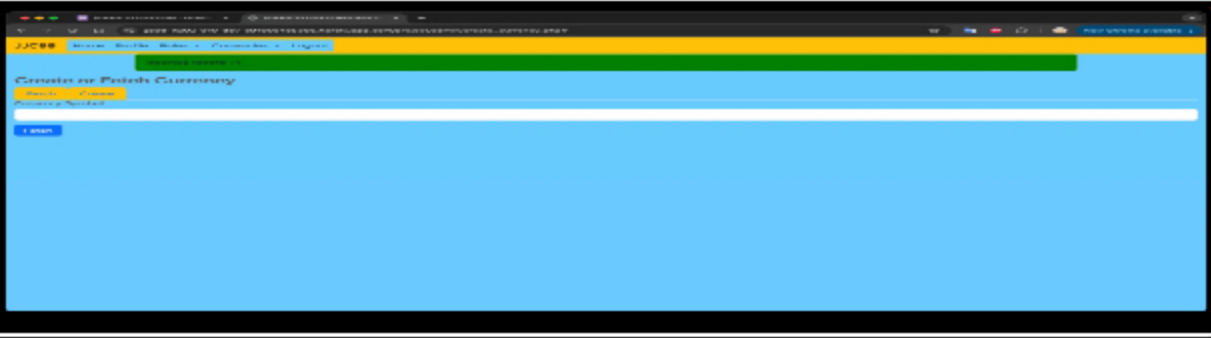
Image Prompt

Weight: *40%*

Details:

- Show the code that handles the data transformation/conversion (from the API)
- Show a Heroku dev example of API getting fetched and loaded into the DB

transform code



fetching an

fetch 14 in the db

Saved: 5/1/2025 8:45:15 PM

Url Prompt

Weight: *20%*

Details:

- Include Heroku prod link to API fetch page
- Include pull request link for this feature

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/admin/create_currency.php
URL
https://jjc88-it202-010-prod-b44e1d

URL #2
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/testApi.php
URL
https://jjc88-it202-010-prod-b44e1d

URL #3
https://github.com/juicejoose/jjc88-it202-011024
URL
https://github.com/juicejoose/jjc88

Saved: 5/1/2025 8:45:15 PM

Text Prompt

Weight: *40%*

Details:

- Will you be using all the data or just a subset?
- Do you need to convert any data or extract pieces of a property's value?
- Briefly explain the transformation/extraction logic to get the API data to the shape your application intends to use

Your Response:

1. I will be using all the data that is given from the api, such as XAU,XAG, and etc
2. Yes the data has to be converted because it is recieved from JSON and also rounded to fit my data.
3. First decode the API response, extract the rates, base Currency, and unit, then format them into key value pairs for a SQL INSERT and maatch to our table structurre.

Saved: 5/1/2025 8:45:15 PM

100%
Task #3 (0.50 pts.) - API Duplicates

Text Prompt

Weight: *25%*
Objective: *API Duplicates*

Details:

- Consider how duplicate entries are handled
  - What happens if you get the same result from the API for an entity that exists but has been unmodified?
  - What happens if you get the same result from the API for an entity that exists but has been modified?

Your Response:

1. It will still insert a new row, even if the data is unchanged. There's no logic to check for existing identical records.
2. It will still insert a new roow with the updaaated values. Older data will remain

Saved: 5/1/2025 8:47:14 PM

100%

Task #4 (0.50 pts.) - API Fetching

Text Prompt

Weight: *25%*
Objective: *API Fetching*

Details:

- Consider how your application will trigger this data fetch
  - (i.e., periodic via cron job, poor man's cron job, on app start/awake, etc)
  - (i.e., User or Admin triggered via a specific page/action)
  - (i.e., Lazy loaded during searches for stale/missing content)
  - What happens if you get the same result from the API for an entity that exists but has been modified?

Your Response:

1. data will be admin triggered through an admin page.
2. A new row will be inserted, keeping modified and unmodified.

Saved: 5/1/2025 8:49:29 PM

98%

Section #2: (1.5 pts.) Data Creation

95%

Task #1 (0.60 pts.) - Validations

Combo Task:

Weight: *40%*
Objective: *Validations*

Image Prompt

Weight: *50%*

Details:

- Show the html validations (code)
- Show the javascript validations (code)
- Show the php validations (code)

html

js

php

Saved: 5/2/2025 1:14:27 PM

Text Prompt

Weight: *50%*

Details:

- Briefly explain the html validations used
- Briefly explain the javascript validation logic
- Briefly explain the php validation logic

Your Response:

1. The HTML form uses the required attribute to ensure certain fields must be fillled before submission and uses input types like number to validate numeric fields
2. Javascript checks that the base currency contains 1-3 letters, the unit contains only letters

100%

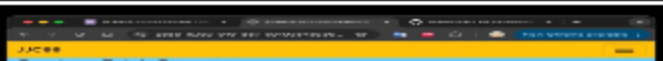
Task #2 (0.60 pts.) - Examples

Image Prompt

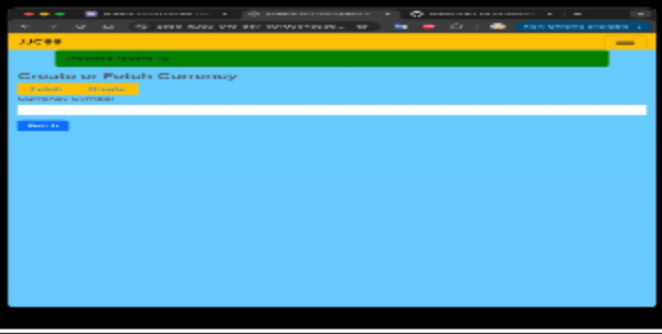
Weight: *40%*
Objective: *Examples*

Details:

- Show a successful creation
- Demonstrate various validation errors
- Ensure heroku url is visible



success 1/2



success 2/2

100%

Task #3 (0.30 pts.) - Links

Url Prompt

Weight: *20%*
Objective: *Links*

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/admin/create_currency.php

URL
https://jjc88-it202-010-prod-b44e1de4(

URL #2
https://github.com/juicejoose/jjc88-it202-[illegible]024

URL
https://github.com/juicejoose/jjc88-it2

Saved: 5/2/2025 12:04:48 PM

Section #3: (1.5 pts.) Data List Page (Many Items)

Task #1 (0.60 pts.) - Code

Combo Task:

Weight: *40%*
Objective: *Code*

Image Prompt

Weight: *50%*

Details:

- Show the code that handles the filter/sort logic
- Show the code that generates the list/grid output
- Styling must be applied

filter/sort logic

list/table code

Saved: 5/1/2025 9:24:37 PM

Text Prompt

Weight: *50%*

Details:

- Briefly explain how the filter/sort logic works and formulates the results
- Briefly explain how the list/grid output is generated
- Briefly note the styling choices

Your Response:

1. The filter/sort logic is a simple if logic to check and match the inputed fields like name and data.
2. Data is fetched usiing query and is passed render table to display them ini a table.
3. Using render functions for table and inputfields, buttons, and colorinig.

Saved: 5/1/2025 9:24:37 PM

100%

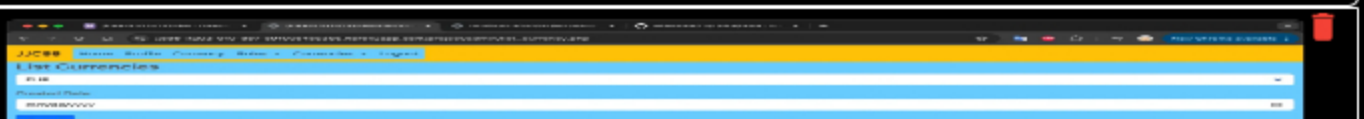
Task #2 (0.60 pts.) - Examples

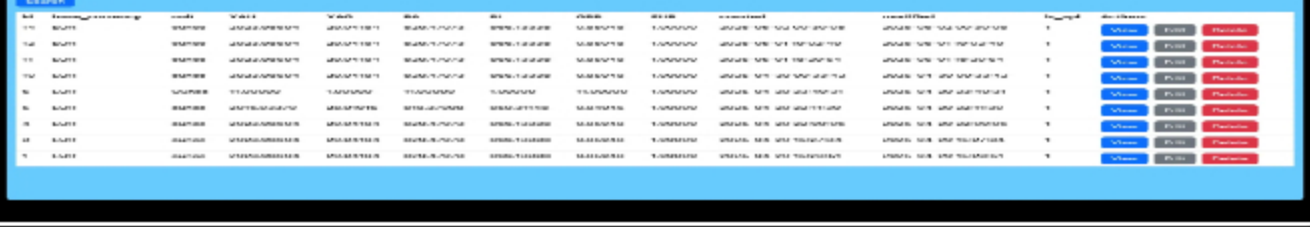
Image Prompt

Weight: *40%*
Objective: *Examples*

Details:

- Show a few variations of filter/sort applied (including no records)
- Each list item should be a summary
- Each list item should have the following links
  - A link to a single view of the specific entity (i.e., a details page)
  - A link to delete this entity (this may be an admin-only functionality, but it should be present for the respective role)
  - A link to edit this entity (this may be an admin-only functionality, but it should be present for the respective role)
- Ensure heroku url is visible





name search

name and date search

100%

Task #3 (0.30 pts.) - Links

Url Prompt

Weight: *20%*
Objective: *Links*

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/admin/list_currency.php

URL
https://jjc88-it202-010-prod-b44e1de4(

URL #2
https://github.com/juicejoose/jjc88-it202-010/pull/24

URL
https://github.com/juicejoose/jjc88-it2

Saved: 5/2/2025 9:58:01 AM

100%

Section #4: (1.5 pts.) View Details Page (Single

Item)

100%

Task #1 (0.60 pts.) - Code

Combo Task:

Weight: *40%*
Objective: *Code*

Image Prompt

Weight: *50%*

Details:

- Show the code that handles loading of the record (and how the id retrieved for usage)
  - Include the code that handles invalid/missing ids
- Show the code that generates the entity output
- Styling must be applied

view code

Saved: 5/2/2025 10:13:35 AM

Text Prompt

Weight: *50%*

Details:

- Briefly explain how the loading logic works
- Briefly explain how the entity output is generated
- Briefly note the styling choices

Your Response:

1. Checks ID from url and if its valid it will run a query on currency and saved in the array.
2. Data is loops in the currency arrray and for each field it will show the name and value
3. For styling used simple container fluid and same infobox and spacinig from other pages.

Saved: 5/2/2025 10:13:35 AM

100%

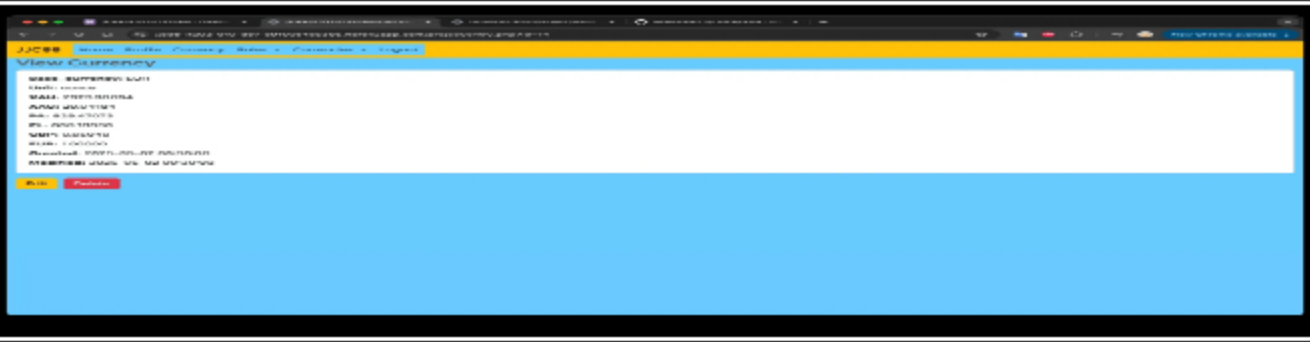
Task #2 (0.60 pts.) - Examples

Image Prompt

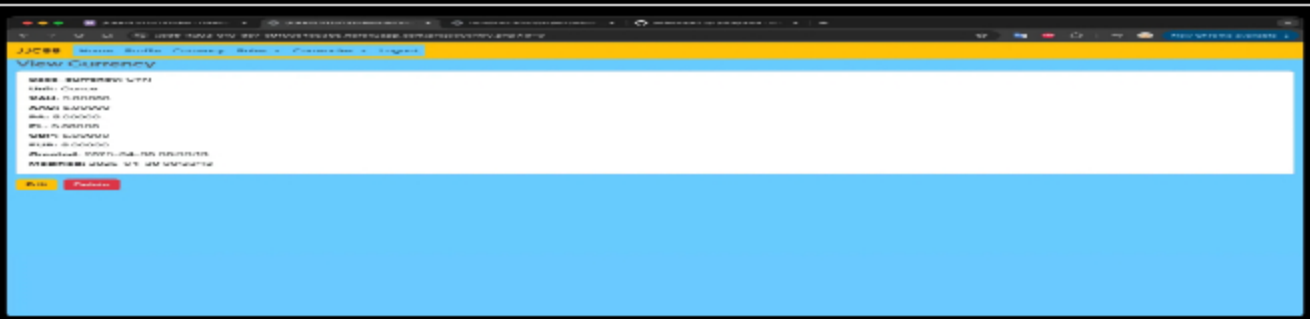
Weight: *40%*
Objective: *Examples*

Details:

- Show a few examples of different entities
- Each entity item should have more details than the summary view (if possible)
- Each entity should have the following links
  - A link to edit the single entity (this may be an admin-only thing, but it should be present for the respective role)
  - A link to delete the single entity (this may be an admin-only thing, but it should be present for the respective role)



entity 1



entity 2

100% Task #3 (0.30 pts.) - Links

Url Prompt

Weight: *20%*
Objective: *Links*

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/entry.php

URL
https://jjc88-it202-010-prod-b44e1de4(

URL #2
https://github.com/juicejoose/jjc88-it202-010/pull/24

URL
https://github.com/juicejoose/jjc88-it2

Saved: 5/2/2025 10:20:11 AM

95% Section #5: (1.5 pts.) Edit Page

100% Task #1 (0.60 pts.) - Code

Combo Task:

Weight: *40%*
Objective: *Code*

Image Prompt

Weight: *50%*

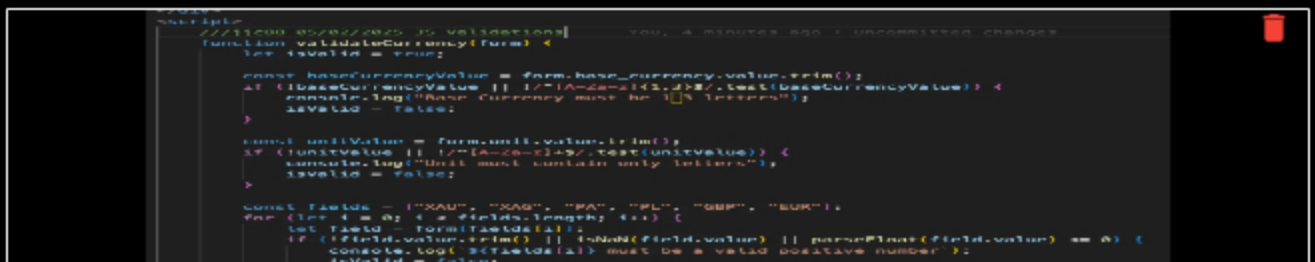
Details:

- Show the code that handles loading of the record (and how the id retrieved for usage)
  - Include the code that handles invalid/missing ids
- Show the code that generates the form with the correct data types and populates it
- Show the html, javascript, and php validation (should be similar to create)
- Styling must be applied

records code

form table code

html val



js val

php val

Saved: 5/2/2025 1:32:23 PM

87%

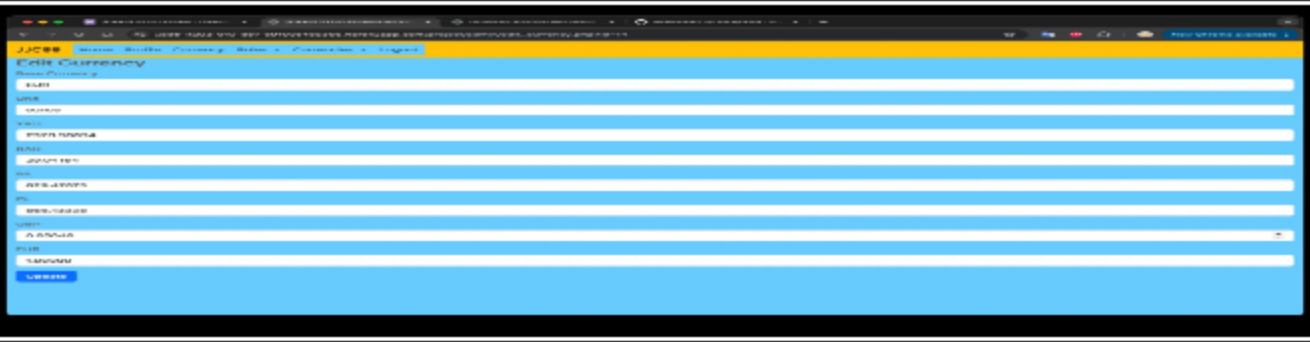
Task #2 (0.60 pts.) - Examples

Image Prompt

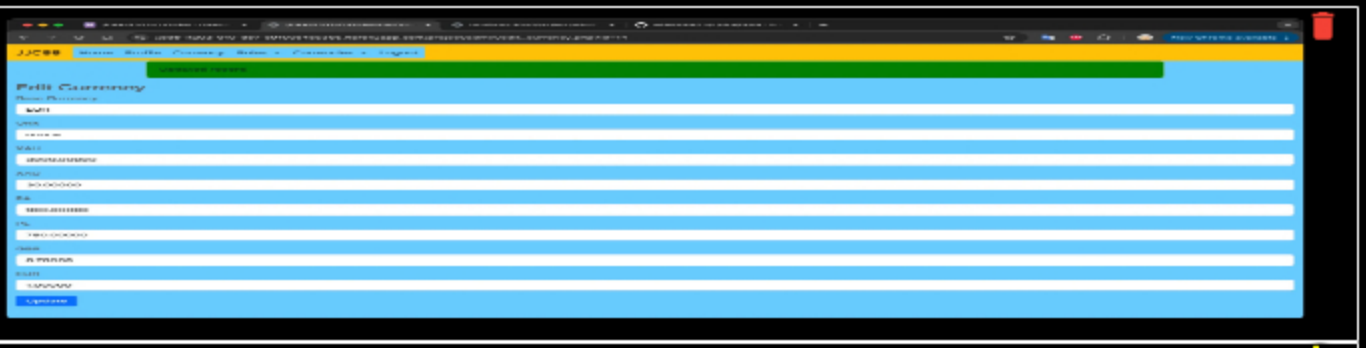
Weight: *40%*
Objective: *Examples*

Details:

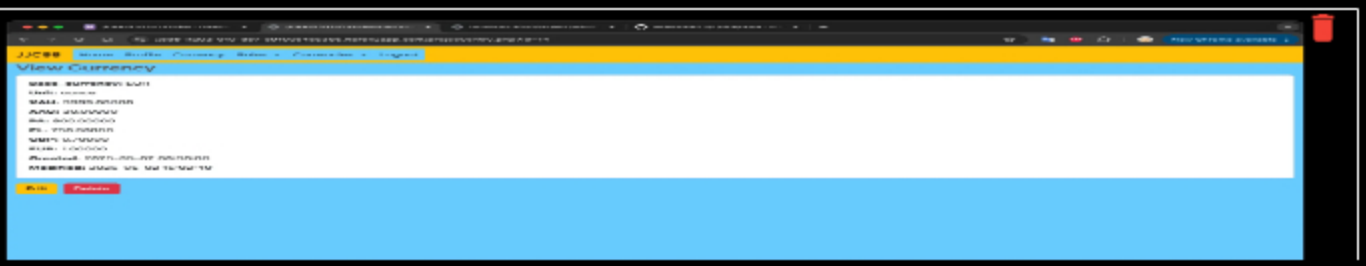
- Show a before and after of updating an entity (clearly caption)
- Successful update should have an appropriate message shown
- The updated data should be shown in the form
- Any errors should have user-friendly messages shown
- Show some examples of validation errors (html, javascript, php)



before edit



after edit



100%

Task #3 (0.30 pts.) - Links

Url Prompt

Weight: *20%*
Objective: *Links*

Details:

- Include the heroku prod link to this page
- Include pull request link for this feature

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/edit_currency.php

URL
https://jjc88-it202-010-prod-b44e1de4(

URL #2
https://github.com/juicejoose/jjc88-it202-010-2024

URL
https://github.com/juicejoose/jjc88-it2

Saved: 5/2/2025 11:05:36 AM

100%

Section #6: (1.5 pts.) Delete Logic

Task #1 (0.60 pts.) - Code

Combo Task:

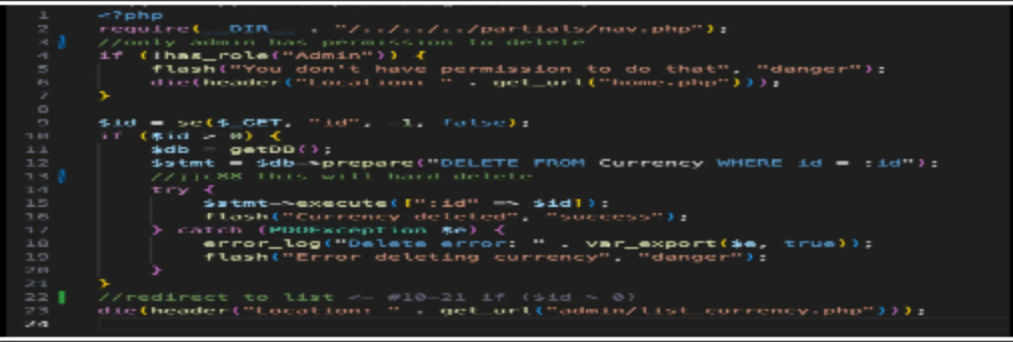
Weight: *40%*
Objective: *Code*

Image Prompt

Weight: *50%*

Details:

- Show the code that handles deletion of the record (and how the id retrieved for usage)
  - Include the code that handles invalid/missing ids
- Handle any necessary roles/permissions (i.e., it's not likely that anyone can delete any entity they choose)
- Show the redirect logic that goes back to the previous page (where the route came from), if it was a list page, it should maintain the filter/sort criteria



```php
<?php
require(__DIR__ . "/../../../partials/nav.php");
//only admin has permission to delete
if (!has_role("Admin")) {
    flash("You don't have permission to do that", "danger");
    die(header("Location: " . get_url("home.php")));
}

$id = se($_GET, "id", -1, false);
if ($id > 0) {
    $db = getDB();
    $stmt = $db->prepare("DELETE FROM Currency WHERE id = :id");
    //!!!XX this will hard delete
    try {
        $stmt->execute([":id" => $id]);
        flash("Currency deleted", "success");
    } catch (PDOException $e) {
        error_log("Delete error: " . var_export($e, true));
        flash("Error deleting currency", "danger");
    }
}
//redirect to list <- #10-21 if ($id > 0)
die(header("Location: " . get_url("admin/list_currency.php")));
```

delete code, permission, and redirect

Saved: 5/2/2025 7:23:51 PM

Text Prompt

Weight: *50%*

Details:

- Briefly explain how the deletion logic works
- Is it a soft or hard delete?
- Note any permissions/rules applied.
- Briefly explain how the redirect maintains the previous route and any filter/sort data

Briefly explain how the redirect maintains the previous route and any filter/sort data

Your Response:

1. It gets the id from the url and runs a delete command if the id is valid
2. This is a hard delette
3. Only admin roles can delete
4. It will redirect back to list but setting will be lost and show all results instead to reset search.

Saved: 5/2/2025 7:23:51 PM

100%

Task #2 (0.60 pts.) - Examples

Image Prompt

Weight: *40%*
Objective: *Examples*

Details:

- Show a successful delete message
- Show a before and after database view of the record being deleted (clearly caption)
- Successful update should have an appropriate message shown
- The updated data should be shown in the form
- Any errors should have user-friendly messages shown

before delete id14

100%

Task #3 (0.30 pts.) - Links

Task #3 (0.30 pts.) - Links

Url Prompt

Weight: *20%*
Objective: *Links*

Details:

- Include the heroku prod link to this page (even though it's not a full page, it's more like how logout works)
- Include pull request link for this feature

URL #1
https://jjc88-it202-010-prod-b44e1de4d69c.herokuapp.com/project/delete_currency.php

URL
https://jjc88-it202-010-prod-b44e1de4(

URL #2
https://github.com/juicejoose/jjc88-it202-010/pull/24

URL
https://github.com/juicejoose/jjc88-it2

Saved: 5/2/2025 11:07:31 AM

100%

Section #7: (0.5 pts.) Misc

100%

Task #1 (0.17 pts.) - Github Details

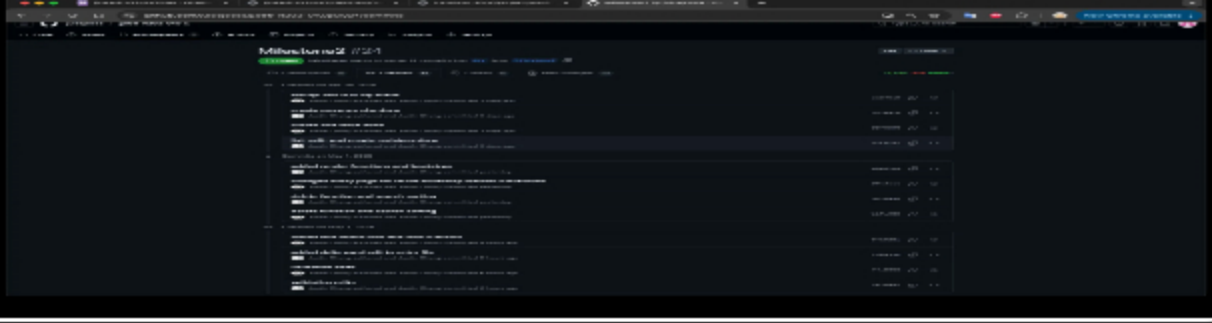
Combo Task:

Weight: *33.33%*
Objective: *Github Details*

Image Prompt

Weight: *60%*

Details:
From the Commits tab of the Pull Request screenshot the commit history



commit tab

Saved: 5/2/2025 7:29:46 PM

Url Prompt

Weight: *40%*

Details:
Include the link to the Pull Request (should end in `/pull/#`)

URL #1
https://github.com/juicejoose/jjc88-it202-011/pull/4

URL
https://github.com/juicejoose/jjc88

Saved: 5/2/2025 7:29:46 PM

100%

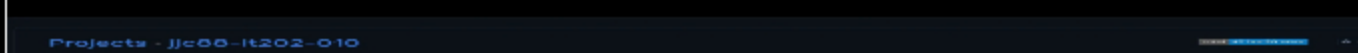
Task #2 (0.17 pts.) - WakaTime - Activity

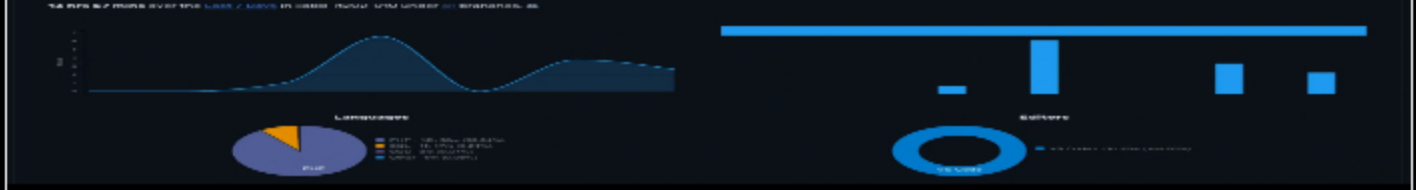
Image Prompt

Weight: *33.33%*
Objective: *WakaTime - Activity*

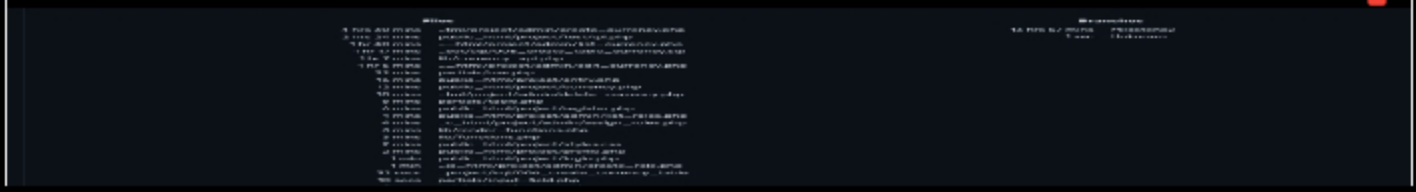
Details:

- Visit the WakaTime.com Dashboard
- Click `Projects` and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary





wakatime



100% Task #3 (0.17 pts.) - Reflection

Weight: *33.33%*
Objective: *Reflection*

Sub-Tasks:

100% Task #1 (0.33 pts.) - What did you learn?

Text Prompt

Weight: *33.33%*
Objective: *What did you learn?*

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

In this milestone2 assignment I learned alot from every concept that was went over. Concepts such as bootstrap, renderfunctions, wrapper files, filtering, and API integration were all present and used heavily in this assignment. Also concepts that were already used are used again which reinforced my skills and knowledge. Concepts such as data mapping and validations are to mention.

Saved: 5/2/2025 11:40:34 AM

100%

Task #2 (0.33 pts.) - What was the easiest part of the assignr

Text Prompt

Weight: *33.33%*
Objective: *What was the easiest part of the assignment?*

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

Easiest part of the assignment was taking screenshots and posting links.

Saved: 5/2/2025 11:33:23 AM

100%

Task #3 (0.33 pts.) - What was the hardest part of the assignı

Text Prompt

Weight: *33.33%*
Objective: *What was the hardest part of the assignment?*

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

I thought the hardest part of the assignment was just getting the code working with the API and understanding new concepts like bootstrap. Although the new concept like bootstrap and renderfunctioons made it more overwhelminig in the beginning, it became alot easier and friendly to use. Also troubleshooting and figuring out validattioons for their respecitiive ddata. Making new files from scratch was also hard such as making delete or view function php files to implement in our list.

Saved: 5/2/2025 7:32:19 PM