Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-007-F2024/it202-api-project-milestone-2-2024-m24/grade/mcp62

Course: IT202-007-F2024
Assigment: [IT202] API Project Milestone 2 2024 M24
Student: Michael P. (mcp62)

Submissions:

Submission Selection

1 Submission [submitted] 12/9/2024 4:33:53 PM

Instructions

COLLAPSE

Implement the Milestone 2 features from the project's proposal document: https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88EWVwfo/view Make sure you add your ucid/date as code comments where code changes are done All code changes should reach the Milestone2 branch Create a pull request from Milestone2 to dev and keep it open until you get the output PDF from this assignment. Gather the evidence of feature completion based on the below tasks. Once finished, get the output PDF and copy/move it to your repository folder on your local machine. Run the necessary git add, commit, and push steps to move it to GitHub Complete the pull request that was opened earlier Create and merge a pull request from dev to prod Upload the same output PDF to Canvas

Branch name: Milestone2

Group

Group: Define Appropriate Tables for Data
Tasks: 2
Points: 1

COLLAPSE

Task

Group: Define Appropriate Tables for Data
Task #1: Screenshots of Table SQL
Weight: ~50%
Points: ~0.50

Points: ~0.50

COLLAPSE

Checklist *The checkboxes are for your own tracking

#	Details
☐ #1	Table(s) should have the 3 core columns we'll always be using (id, created, modified) plus additional columns for the incoming API data
☐ #2	Columns should be logical and thought out (not valid to have a single field of JSON data or similar

Sub-Task

100%

Group: Define Appropriate Tables for Data
Task #1: Screenshots of Table SQL
Sub Task #1: Screenshot of the table (you can duplicate this subtask as needed)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

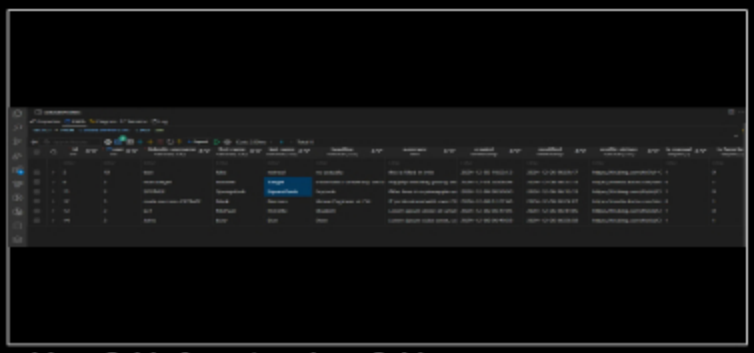


table w fields for api, and req fields

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain the columns and what data they represent (briefly), also note any normalization that may have been necessary

Response:

ID is a unique value given to each row, user_id is given dependent on what the users ID is, linkedin username represents username on profile (linkedin), first_name represents profiles name from feild, headline is another api field scraped from linkedin, summary is the summary context from linkedin profiles (api), created timestamp, last modified timestamp, profile picture is the image url link, (150x150 profile picture on linkedin scraped from api), is manual (0 false, 1 true), is_favorite(0 false, 1 true).

End of Task 1

Task

100%

Group: Define Appropriate Tables for Data
Task #2: Add the pull request link for the branch related to this feature
Weight: ~50%
Points: ~0.50

COLLAPSE

Details:
Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature.

Task URLs

URL #1
https://github.com/Onervv/mcp62-IT202-007/pull/42

URL
https://github.com/Onervv/mcp62-IT202-007/pul

End of Task 2

End of Group: Define Appropriate Tables for Data
Task Status: 2/2

Group

100%

Group: Data Creation Page
Tasks: 4
Points: 2

COLLAPSE

Task

100%

Group: Data Creation Page
Task #1: Screenshots of the creation page
Weight: ~25%
Points: ~0.50

COLLAPSE

Details:
Heroku dev url must be visible in all relevant screenshots

Columns: 1

Sub-Task

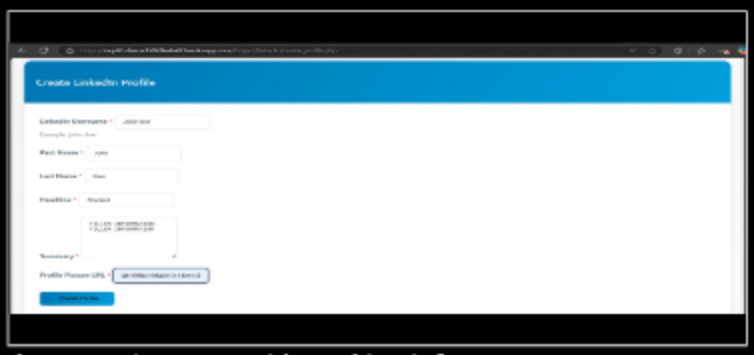
100%

Group: Data Creation Page
Task #1: Screenshots of the creation page
Sub Task #1: Show potentially valid data filled in for the custom creation page

Task Screenshots

Gallery Style: 2 Columns

4 2 1



data creation page with working info

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

Group: Data Creation Page
Task #1: Screenshots of the creation page
Sub Task #2: Show how the API data is fetched for API data (must be server-side)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Api fetching settings

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the code

Response:

The API call is being made using jQuery's $.ajax() method in the browser. After receiving the data, it's then sent to store_profile.php to be saved in the database

Sub-Task 100%

Group: Data Creation Page
Task #1: Screenshots of the creation page
Sub Task #3: Show examples of validation messages

Task Screenshots

Gallery Style: 2 Columns

4 2 1

client side validation

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

Group: Data Creation Page
Task #1: Screenshots of the creation page
Sub Task #4: Show an example of successful creation message

Task Screenshots

Gallery Style: 2 Columns

4 2 1

properly stored and shows validation message

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

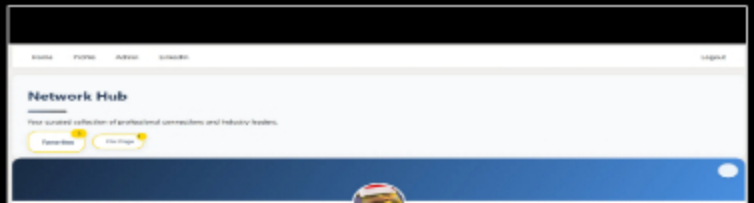
Sub-Task 100%

Group: Data Creation Page
Task #1: Screenshots of the creation page
Sub Task #5: Design/Style should be considered (i.e., bootstrap, custom css, etc)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



shows off custom css using bootstrap framework

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain your design choices

Response:

I wanted the theme to look similar to linkedin considering thats the information the user is going to be handling in this webpage, I wanted it to be clear and easy to view and I wanted there to be nice animations and effects scattered across which there are many subtle effects.

End of Task 1

Task

100%

Group: Data Creation Page
Task #2: Screenshots of creation page code
Weight: ~25%
Points: ~0.50

COLLAPSE

Details:
Include ucid/date comments for each code screenshot

Columns: 1

Sub-Task

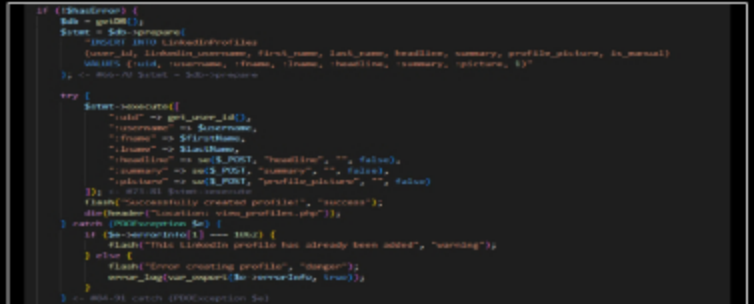
100%

Group: Data Creation Page
Task #2: Screenshots of creation page code
Sub Task #1: Form should have correct data types for each property being requested

Task Screenshots

Gallery Style: 2 Columns

4 2 1



primary creation page code, creates new profiles checks for validation and duplicates

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly talk about each field type and why it was chosen

Response:

the code is for manually creating LinkedIn profiles (note is_manual = 1 in the INSERT) It uses prepared statements to prevent SQL injection The se() function is a helper that safely extracts POST data

Sub-Task

100%

Group: Data Creation Page

Task #2: Screenshots of creation page code

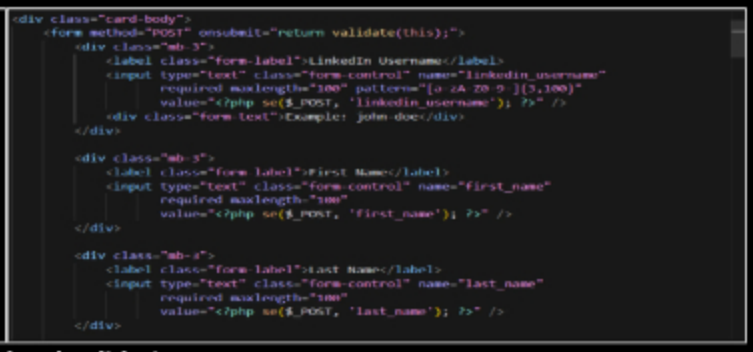
Sub Task #2: Form should have correct validation for each field (HTML, JS, and PHP)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

PHP



html validation

JS validation

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the validations

Response:

JS - Validates LinkedIn username format (letters, numbers, hyphens, 3-100 chars). Looks for empty required fields

(first name, last name), Validates profile picture URL format if provided, Returns false to prevent form submission if any validation fails.

PHP- Validates required fields (username, first/last name). Checks LinkedIn username format with regex. Uses validateProfilePictureUrl() function for URL validation. Sets $hasError flag if any validation fails.

Sub-Task

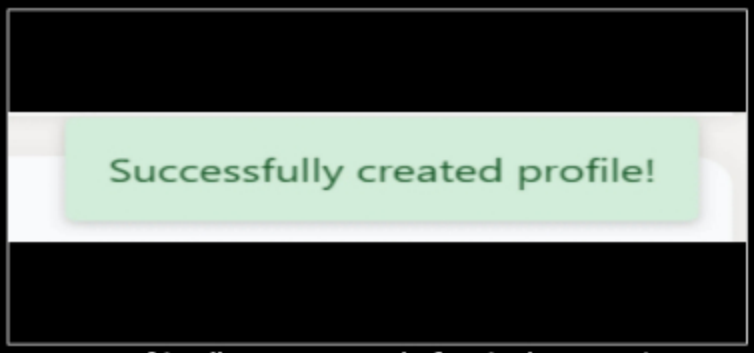
100%

Group: Data Creation Page
Task #2: Screenshots of creation page code
Sub Task #3: Successful creation should have a user-friendly message

Task Screenshots

Gallery Style: 2 Columns

4 2 1



same user friendly message as before in data creation message confirmation

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain how duplicate/existing data is handled
Response:

Duplicate data is handled is a try block, if certain values are re inputed a flash message is sent to tell the user the profile aalready exists

Sub-Task

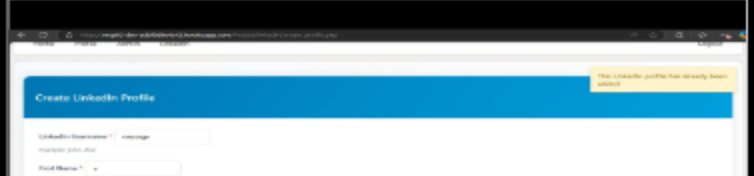
100%

Group: Data Creation Page
Task #2: Screenshots of creation page code
Sub Task #4: Any errors should have user-friendly messages

Task Screenshots

Gallery Style: 2 Columns

4 2 1



example of user-friendly error on data creation page, additional messages exist for incorrect field information

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly describe the scenarios

Response:

1. if the password doesnt meet requirements
2. if the username doesnt meet requirements
3. if the data already exists
4. other uncaught errors throw a message
5. if your not logged in you dont have access to the page
6. if you does fill out certain data fields on creation page

Sub-Task

100%

Group: Data Creation Page

Task #2: Screenshots of creation page code

Sub Task #5: Include the form/process for fetching API data

Task Screenshots

Gallery Style: 2 Columns

4 2 1

```
try {
    $stmt->execute([
        ":uid" => get_user_id(),
        ":username" => $username,
        ":fname" => $firstName,
        ":lname" => $lastName,
        ":headline" => se($_POST, "headline", "", false),
        ":summary" => se($_POST, "summary", "", false),
        ":picture" => se($_POST, "profile_picture", "", false)
    ]); <- #73-81 $stmt->execute
```

local create feature

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the steps (include how duplicates are handled)

Response:

Locally creates user filled data into db fields, does not need to fetch api for locally created data

Sub-Task

Group: Data Creation Page

100%

Task #2: Screenshots of creation page code

Sub Task #6: Include some indicator between custom data and API data

Task Screenshots

Gallery Style: 2 Columns

4 2 1

is_manual indicator

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly mention what the indicator is (i.e., api_id if the API has ids or is_api as a boolean-like column, etc)

Response:

boolean column checks if the profile was created by inputed values, is_manual == 1 (true) else, data takes the value 0 (indicating the data must have been fetched from api)

Sub-Task

100%

Group: Data Creation Page

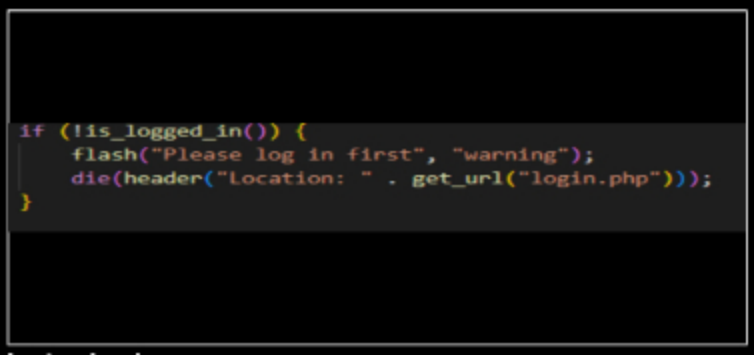
Task #2: Screenshots of creation page code

Sub Task #7: Include any other rules like role guards and login checks

Task Screenshots

Gallery Style: 2 Columns

4 2 1



login check

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the logic/reasoning

Response:

login check, ensures a user is logged in for proper user mapping. No roles needed as any user at any permission can create custom profiles, which will be isolated to there account using their user_id

End of Task 2

Task

100%

Group: Data Creation Page
Task #3: Screenshot of records from DB
Weight: ~25%
Points: ~0.50

COLLAPSE

Columns: 1

Sub-Task

100%

Group: Data Creation Page
Task #3: Screenshot of records from DB
Sub Task #1: Show at least one record fetched from the API

Task Screenshots

Gallery Style: 2 Columns

4 2 1

shows record matttoegel fetched from api (is manual 0)

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown (note what differs from a custom record)*

Sub-Task

100%

Group: Data Creation Page
Task #3: Screenshot of records from DB
Sub Task #2: Show at least one record created via the creation form

Task Screenshots

Gallery Style: 2 Columns

4 2 1

is_manual 1 (test is created data)

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown (note what differs from the API record)*

End of Task 3

Task

100%

Group: Data Creation Page
Task #4: Add related links
Weight: ~25%
Points: ~0.50

COLLAPSE

Checklist

*The checkboxes are for your own tracking

#	Details
☐ #1	Include the heroku prod link for this page
☐ #2	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

Task URLs

URL #1
https://mcp62-prod-c0d770baface.herokuapp.com/Project/linkedin/create_profile.php

URL
https://mcp62-prod-c0d770baface.herokuapp.co

URL #2
https://github.com/Onervv/mcp62-IT202-007/pull/46

URL
https://github.com/Onervv/mcp62-IT202-007/pul

End of Task 4

End of Group: Data Creation Page
Task Status: 4/4

Group

100%

Group: Data List Page (many entities)
Tasks: 3
Points: 2

COLLAPSE

Task

100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Weight: ~33%
Points: ~0.67

COLLAPSE

Details:
Heroku dev url must be visible in all relevant screenshots

Columns: 1

Sub-Task

100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #1: Show the page of your entities listed (have a reasonable number shown)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

zoomed out to see many entities

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly describe the page

Response:

data list page shows manually created and fetch linkedin profiles that included specific fields

Sub-Task

100%

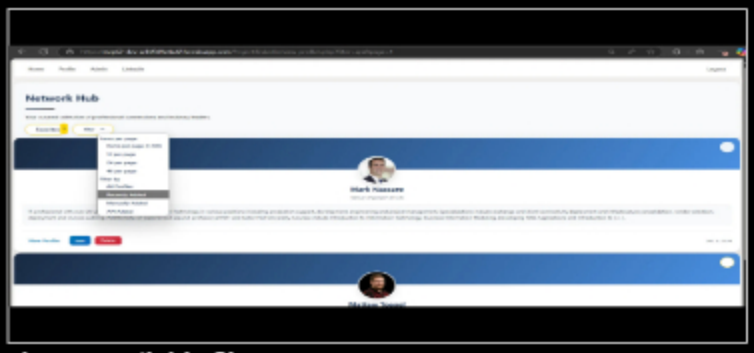
Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #2: Show the filter/sort form based on your data and the required limit field

Task Screenshots

Gallery Style: 2 Columns

Gallery Style: 2 Columns

4 2 1



shows available filters

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly mention what's available to the user

Response:

You can filter by most recently modified (not creation), api created, manually created, and with no filters, additionally there are options for pagination

Sub-Task 100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #3: Demonstrate a few varied filters/sorts

Task Screenshots

Gallery Style: 2 Columns

4 2 1

manually added

api filter

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #4: Demonstrate a filter that doesn't have any records (should show an appropriate message)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

no favorites filter message

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #5: Each list item should have a link of single view (i.e., details), edit, and delete (some of which may only be visible to admin users, include examples from different user roles)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

edit and delete, view link and favorite button on each profile

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Mention which users can interact with the view, edit, and delete links

Response:

admins have access to all functionality, none admins cannot edit or delete profiles, but can still favorite

Sub-Task 100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #6: Each list item should have a summary of the entity (likely won't be the entire entity data)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

summary field

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

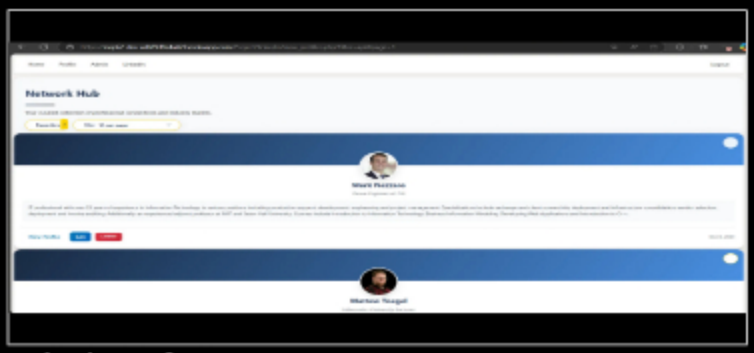
100%

Group: Data List Page (many entities)
Task #1: Screenshots of the list page
Sub Task #7: Design/Style should be considered (i.e., bootstrap, custom css, etc)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



style shown for page

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain your design choices

Response:

The css shows useful information like how many users you wanna see on each page, additionally buttons are resposive and give indications when clicked. The style again was to maintain a modern white and blue linkedin style.

End of Task 1

Task

Group: Data List Page (many entities)
Task #2: Screenshots of the list page code

100%

Weight: ~33%

Points: ~0.67

COLLAPSE

Details:

Include ucid/date comments for each code screenshot

Columns: 1

Sub-Task

100%

Group: Data List Page (many entities)

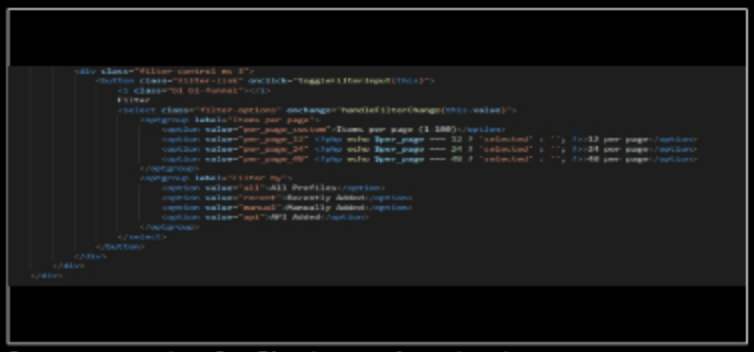
Task #2: Screenshots of the list page code

Sub Task #1: Show the filter/sort form generation

Task Screenshots

Gallery Style: 2 Columns

4 2 1



form generation for filtering and pagination

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain how the code works (i.e, some stuff may be dynamic)

Response:

on click it toggles the values contained within the drop down, what is then displayed is handled by the filteirng and pagination options some of whcih are daata driven and others such as the pagination which are user input driven

Sub-Task

100%

Group: Data List Page (many entities)

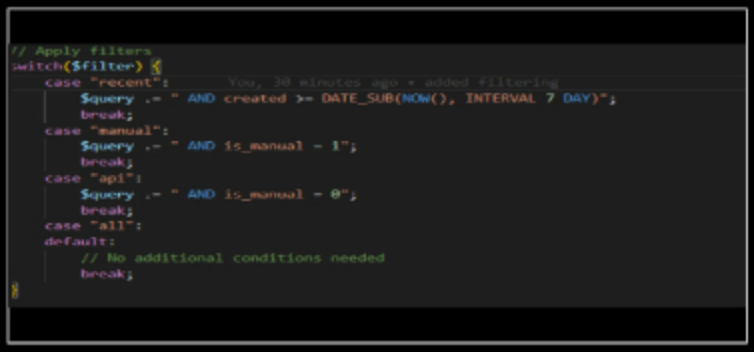
Task #2: Screenshots of the list page code

Sub Task #2: Show the DB query and how the filter/sort is handled (including the restriction on the limit field)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```php
// Apply filters
switch($filter) {
    case "recent":
        $query .= " AND created >= DATE_SUB(NOW(), INTERVAL 7 DAY)";
        break;
    case "manual":
        $query .= " AND is_manual = 1";
        break;
    case "api":
        $query .= " AND is_manual = 0";
        break;
    case "all":
    default:
        // No additional conditions needed
        break;
}
```

DB query

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the related logic

Response:

link allows for seperation o

Sub-Task

100%

Group: Data List Page (many entities)

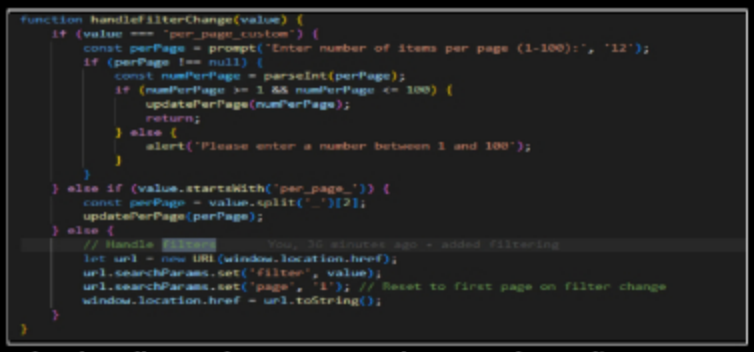
Task #2: Screenshots of the list page code

Sub Task #3: Show how the output is generated and displayed

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```javascript
function handleFilterChange(value) {
    if (value === 'per_page_custom') {
        const perPage = prompt('Enter number of items per page (1-100):', '12');
        if (perPage !== null) {
            const numPerPage = parseInt(perPage);
            if (numPerPage >= 1 && numPerPage <= 100) {
                updatePerPage(numPerPage);
                return;
            } else {
                alert('Please enter a number between 1 and 100');
            }
        }
    } else if (value.startsWith('per_page_')) {
        const perPage = value.split('_')[2];
        updatePerPage(perPage);
    } else {
        // Handle filter
        let url = new URL(window.location.href);
        url.searchParams.set('filter', value);
        url.searchParams.set('page', '1'); // Reset to first page on filter change
        window.location.href = url.toString();
    }
}
```

What handles and regenerates the page depending on selected filters

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the related logic

Response:

The function manages two main tasks: handling pagination (letting users choose how many profiles to show per page, either through preset options or a custom number) and filtering (allowing users to view specific types of profiles). When either option changes, it updates the URL and refreshes the page to show the desired results.

Sub-Task

100%

Group: Data List Page (many entities)
Task #2: Screenshots of the list page code
Sub Task #4: Show any restrictions like role guard or login checks

Task Screenshots

Gallery Style: 2 Columns

4 2 1

```php
<?php
require(__DIR__ . "/../../../partials/nav.php");
if (!is_logged_in()) {
    flash("Please log in first", "warning");
    die(header("Location: " . get_url("login.php")));
```

only role guard

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the logic/reasoning

Response:

only role guard that exists for this feature is making sure the user is logged in for obvious reasoning.

End of Task 2

Task

100%

Group: Data List Page (many entities)
Task #3: Add related links
Weight: ~33%
Points: ~0.67

COLLAPSE

Checklist

*The checkboxes are for your own tracking

#	Details
#1	Include the heroku prod link for this page
#2	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

Task URLs

URL #1

https://github.com/Onervv/mcp62-IT202-007/pull/51

URL

https://github.com/Onervv/mcp62-IT202-007/pul

URL #2
https://mcp62-prod-adbf9d9e4a42.herokuapp.com/Project/linkedin/view_profiles.php?filter=api&page=1

URL
https://mcp62-prod-adbf9d9e4a42.herokuapp.co

End of Task 3

End of Group: Data List Page (many entities)
Task Status: 3/3

Group

100%

Group: View Details Page (single entity)
Tasks: 3
Points: 1

COLLAPSE

Task

100%

Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Weight: ~33%
Points: ~0.33

COLLAPSE

Details:
Heroku dev url must be visible in all relevant screenshots

Columns: 1

Sub-Task

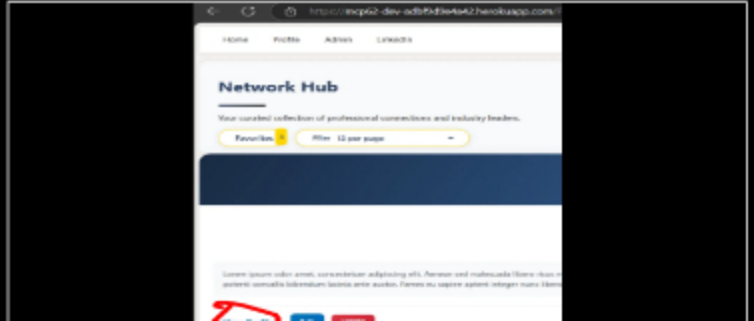
100%

Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Sub Task #1: Entity should be fetch by id (via the url)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



View Profile

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

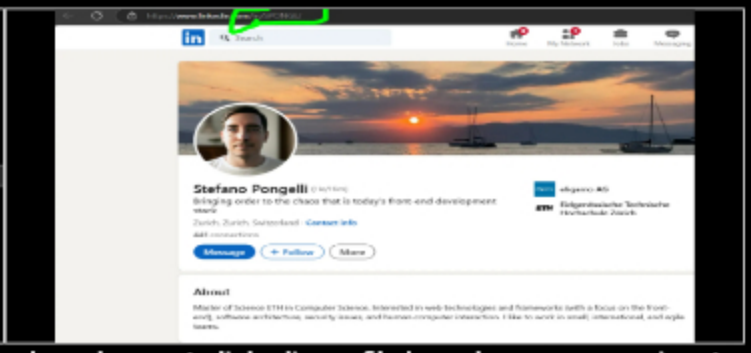
Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Sub Task #2: A missing id should redirect back to the list page with an applicable message

Task Screenshots

Gallery Style: 2 Columns

4 2 1

url example



url sends user to linkedin profile based on username input field

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Sub Task #3: Design/Style should be considered (i.e., bootstrap, custom css, etc)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

full view user profile

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain your design choices

Response:

given the context of my project since im selecting only a few values from a large portion of fetchable infomation from my api, it doesnt make sense to make the full view profile card, instead the code maps each user to the linkedin url to look at the person profile fully if needed.

Sub-Task

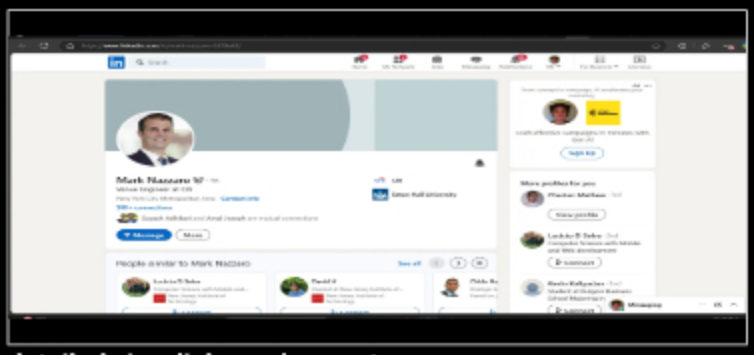
100%

Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Sub Task #4: Data shown should be more detailed/inclusive than the summary view

Task Screenshots

Gallery Style: 2 Columns

4 2 1



detailed view link sends user to

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the details shown and how it differs from the list view

Response:

list view and details view are combine in this instance the full amount of information is presented on each users card, in the case if I were to use more data fields being fetch from the api that could work, however already there is a large amount of potential data for each user given the summary alone. I think the design idea of this project is to specificaly reduce the over-bearing amount of information linkedin pushes onto its user (as linkedin is a social media platform for one) there is alot of content people can get dragged into when another larger portion of what linkedin is for is networking and job hunting, that is what I want to target.

Sub-Task

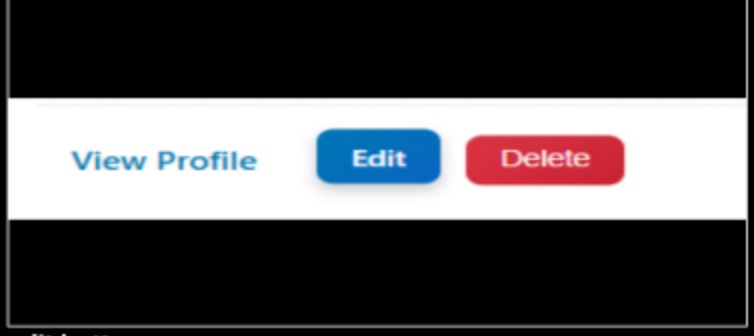
100%

Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Sub Task #5: There should be a link to edit the entity (this may be an admin-only thing, but it should be present for the respective role)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



edit button

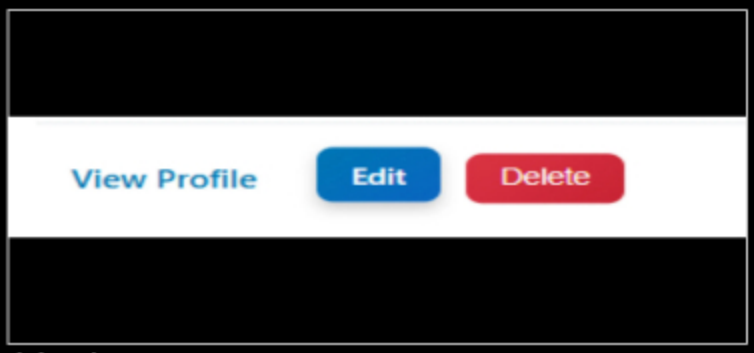
Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

Sub-Task
100%
Group: View Details Page (single entity)
Task #1: Screenshots of the details page
Sub Task #6: There should be a link to delete the entity (this may be an admin-only thing, but it should be present for the respective role)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



delete button

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

End of Task 1

Task
100%
Group: View Details Page (single entity)
Task #2: Screenshots of the details page code
Weight: ~33%
Points: ~0.33
COLLAPSE

Details:
Include ucid/date comments for each code screenshot

Columns: 1

Sub-Task

100%

Group: View Details Page (single entity)
Task #2: Screenshots of the details page code
Sub Task #1: Show how id is fetched

Task Screenshots

Gallery Style: 2 Columns

4 2 1

id fetched

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the logic and how it's handled when the property is missing or not "valid"

Response:

logic is directly in the html for this "feature", if the user profile does not exist then a 404 error is shown when searching for linkedin profiles users page (this is improtant becasue it make sure the user is putting in valid infomration to the edit field. The code functions by providing details of the linkedin url up to the point of the username, that is then check based on the card view profile information ($profile, "Linkedin_username")

Sub-Task

100%

Group: View Details Page (single entity)
Task #2: Screenshots of the details page code
Sub Task #2: Show the DB query to get the record

Task Screenshots

Gallery Style: 2 Columns

4 2 1

query not needed for implementation

query not needed for implementation

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the logic

Response:

Again, logic simply sends user to linkedin profile page, if additionally fields where extracted from my api it might make sense for a webpage details view but my primary view functions as the detailed view at this time

Sub-Task

100%

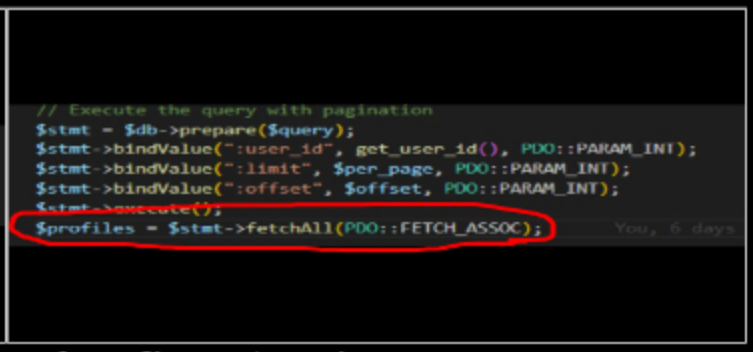
Group: View Details Page (single entity)
Task #2: Screenshots of the details page code
Sub Task #3: Show the code related to presenting the data and showing the links

Task Screenshots

Gallery Style: 2 Columns

4 2 1

code for link



Var for profile mapping code

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the logic

Response:

$profiles i fetched by association within the db, then mapped to href in html

End of Task 2

Task

100%

Group: View Details Page (single entity)
Task #3: Add related links
Weight: ~33%
Points: ~0.33

COLLAPSE

Checklist

*The checkboxes are for your own tracking

#	Details
☐ #1	Include the heroku prod link for this page
☐ #2	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

Task URLs

URL #1
https://github.com/Onervv/mcp62-IT202-007/pull/46

URL
https://github.com/Onervv/mcp62-IT202-007/pul

URL #2
https://mcp62-prod-adbf9d9e4a42.herokuapp.com/Project/linkedin/view_profiles.php

URL
https://mcp62-prod-adbf9d9e4a42.herokuapp.co

End of Task 3

End of Group: View Details Page (single entity)
Task Status: 3/3

Group

100%

Group: Edit Data Page
Tasks: 4
Points: 2

COLLAPSE

Task

100%

Group: Edit Data Page
Task #1: Screenshots of the edit page
Weight: ~25%
Points: ~0.50

COLLAPSE

Details:
Heroku dev url must be visible in all relevant screenshots

Columns: 1

Sub-Task

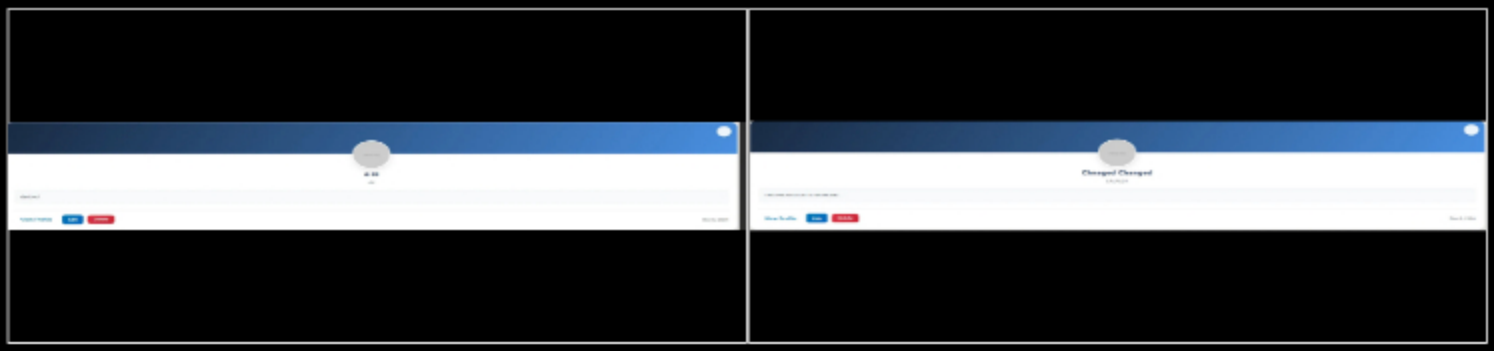
100%

Group: Edit Data Page
Task #1: Screenshots of the edit page
Sub Task #1: Show before and after screenshots of data you'll edit (two screenshots required)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



before after

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain what you changed

Response:

I changed the first name, last name, header and summary

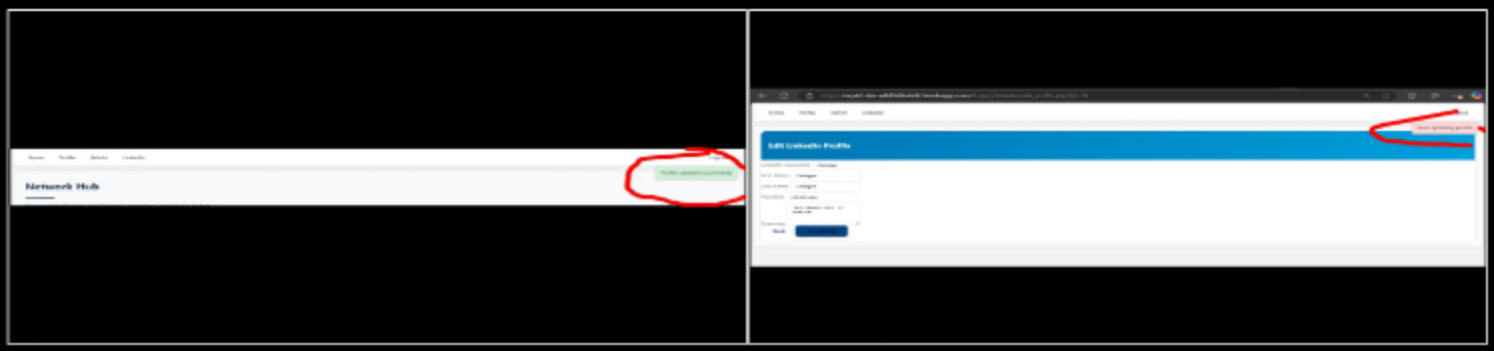
Sub-Task 100%

Group: Edit Data Page
Task #1: Screenshots of the edit page
Sub Task #2: Show examples of validation messages

Task Screenshots

Gallery Style: 2 Columns

4 2 1



validation message error handling

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

Group: Edit Data Page
Task #1: Screenshots of the edit page
Sub Task #3: Show an example of successful edit messages

Task Screenshots

Gallery Style: 2 Columns

4 2 1

sucsessful edit message

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Edit Data Page

Task #1: Screenshots of the edit page

Sub Task #4: Design/Style should be considered (i.e., bootstrap, custom css, etc)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

stlye of edit page

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain your design choices

Response:

Still wanted it to look similar to linkedin, clean modern look, the main intentions where to create a ui that was simple for the user to navitigate through.

End of Task 1

Task

100%

Group: Edit Data Page

Task #2: Screenshots of edit page code

Weight: ~25%

Points: ~0.50

COLLAPSE

Details:
Include ucid/date comments for each code screenshot

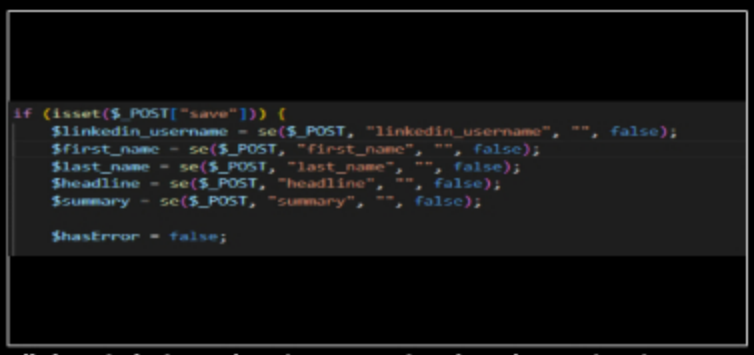
Columns: 1

Sub-Task
100%
Group: Edit Data Page
Task #2: Screenshots of edit page code
Sub Task #1: Form should have correct data types for each property being requested

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```php
if (isset($_POST["save"])) {
    $linkedin_username = se($_POST, "linkedin_username", "", false);
    $first_name = se($_POST, "first_name", "", false);
    $last_name = se($_POST, "last_name", "", false);
    $headline = se($_POST, "headline", "", false);
    $summary = se($_POST, "summary", "", false);

    $hasError = false;
```

all data is being taken in as a string (varchar & tiny_int are the only types for sql table used) and checked using safe echo

Caption(s) (required) ✓
Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the data type choices
Response:

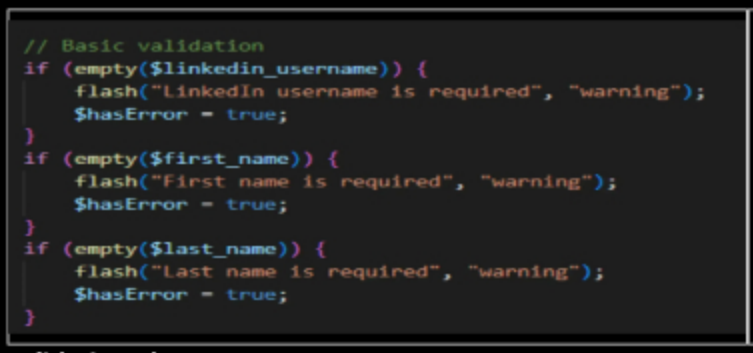
gets values from post array, checks if the key doesnt exisits within post array using linkedin username it returns an empty string to easily be handled and makes sure special character are escaped to prevent issues, vars are then used later to check if they can be updated

Sub-Task
100%
Group: Edit Data Page
Task #2: Screenshots of edit page code
Sub Task #2: Form should have correct validation for each field (HTML, JS, and PHP)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```php
// Basic validation
if (empty($linkedin_username)) {
    flash("LinkedIn username is required", "warning");
    $hasError = true;
}
if (empty($first_name)) {
    flash("First name is required", "warning");
    $hasError = true;
}
if (empty($last_name)) {
    flash("Last name is required", "warning");
    $hasError = true;
}
```

validation php

```js
<script>
function validate(form) {
    let isValid = true;

    // Basic validation
    if (!form.linkedin_username.value.trim()) {
        flash("LinkedIn username is required", "warning");
        isValid = false;
    }
    if (!form.first_name.value.trim()) {
        flash("First name is required", "warning");
        isValid = false;
    }
    if (!form.last_name.value.trim()) {
        flash("Last name is required", "warning");
        isValid = false;
    }

    return isValid;
}
</script>
```

validation js

```php
<div class="card body">
    <form method="POST" onsubmit="return validate(this);">
        <div class="mb-3">
            <label for="linkedin_username" class="form-label">LinkedIn Username</label>
            <input type="text"
                class="form-control"
                id="linkedin_username"
                name="linkedin_username"
                value="<?php se($profile, 'linkedin_username'); ?>"
                required>
        </div>
```

html validation

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the validations

Response:

php validation checks fields server side making sure required fields are filled. JS validation does the same but checks the client side form prior to prior to php validation. HTML validation check on form submission additionally for page issues such as invalid inputed details.

Sub-Task

100%

Group: Edit Data Page

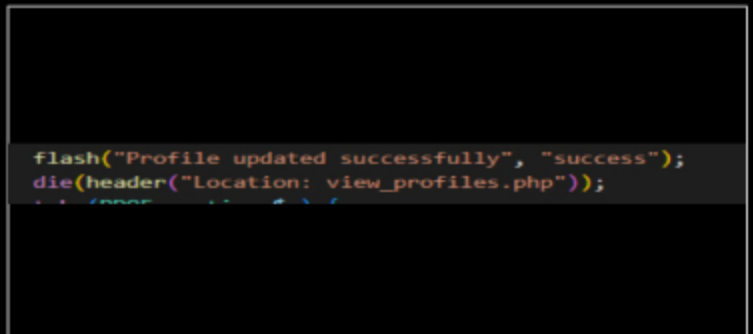
Task #2: Screenshots of edit page code

Sub Task #3: Successful edit should have a user-friendly message

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```php
flash("Profile updated successfully", "success");
die(header("Location: view_profiles.php"));
```

user friendly message code

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Provide a high-level step-by-step of how the fetching of the record, populating the form, and the update works and gets changed in your DB

Response:

The edit profile process begins with a security check, making sure only Admin users can access the page and validating that a proper profile ID was used. Once these checks pass, the system fetches the existing profile data from the LinkedInProfiles database table using the provided ID. The fetched profile data is then used to populate a form, displaying the current values for fields like LinkedIn username, first name, last name, headline, and summary. Users can see and modify the existing information directly in the form fields. When the form is submitted, the system captures all the updated values from the POST request and runs them through validation checks. These checks ensure required fields like LinkedIn username, first name, and last name aren't empty. If all validation passes, the system prepares and executes an SQL UPDATE statement to modify the profile record in the database. The statement updates all fields with their new values for the specific profile ID. Upon successful update, the user receives a success message and is redirected back to the profiles list page. If any errors occur during this process, appropriate error messages are displayed and logged.

Sub-Task

100%

Group: Edit Data Page

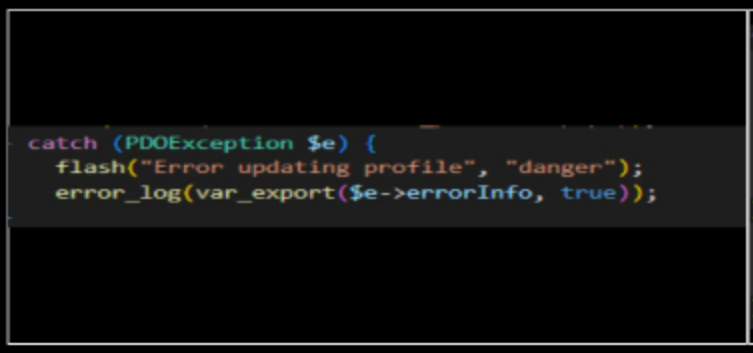
Task #2: Screenshots of edit page code

Sub Task #4: Any errors should have user-friendly messages

Task Screenshots

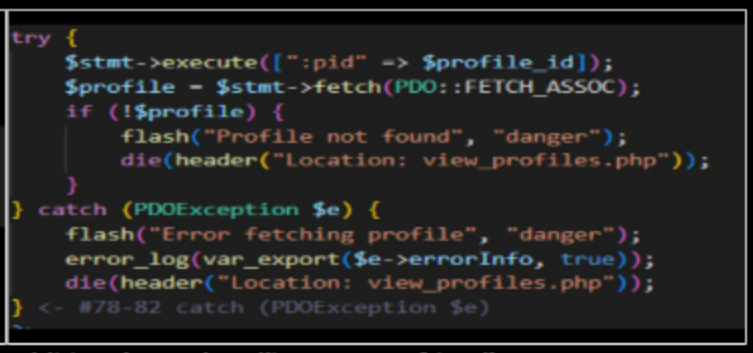
Gallery Style: 2 Columns

4 2 1



```
catch (PDOException $e) {
    flash("Error updating profile", "danger");
    error_log(var_export($e->errorInfo, true));
```

error message



```
try {
    $stmt->execute([":pid" => $profile_id]);
    $profile = $stmt->fetch(PDO::FETCH_ASSOC);
    if (!$profile) {
        flash("Profile not found", "danger");
        die(header("Location: view_profiles.php"));
    }
} catch (PDOException $e) {
    flash("Error fetching profile", "danger");
    error_log(var_export($e->errorInfo, true));
    die(header("Location: view_profiles.php"));
} <- #78-82 catch (PDOException $e)
```

additional error handling w/ user friendly messages

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the possible errors

Response:

Shows a user-friendly message using the flash() function Logs technical details when appropriate (for database errors) Redirects users to an appropriate page when necessary Uses appropriate severity levels ("warning" or "danger") Prevents further execution when critical errors occur (using die())

Sub-Task

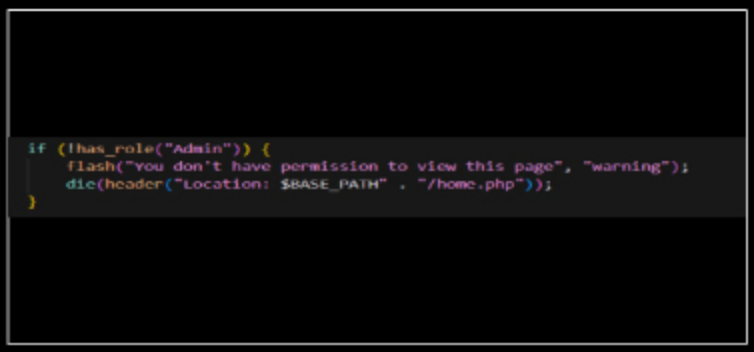
100%

Group: Edit Data Page
Task #2: Screenshots of edit page code
Sub Task #5: Include any other rules like role guards and login checks

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: $BASE_PATH" . "/home.php"));
}
```

admin role guard

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain the logic/reasoning

Response:

ensures only admin can edit profiles, made this logical descision so for further implementations protecting the quality of custom user-inputed profiles will be maintained.

End of Task 2

Task

100%

Group: Edit Data Page
Task #3: Screenshot of records from DB
Weight: ~25%
Points: ~0.50

COLLAPSE

Sub-Task

100%

Group: Edit Data Page
Task #3: Screenshot of records from DB
Sub Task #1: Show a before and after screenshot of the record (two screenshots)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

before after

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain what differs

Response:

I changed the first and last name aswell as the headline to be gary the snail based information

End of Task 3

Task

100%

Group: Edit Data Page
Task #4: Add related links
Weight: ~25%
Points: ~0.50

COLLAPSE

Checklist *The checkboxes are for your own tracking

#	Details
☐ #1	Include the heroku prod link for this page
☐ #2	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

Task URLs

URL #1
https://github.com/Onervv/mcp62-IT202-007/pull/46

URL
https://github.com/Onervv/mcp62-IT202-007/pul

URL #2
https://mcp62-prod-adbf9d9e4a42.herokuapp.com/Project/linkedin/edit_profile.php?id=11

URL
https://mcp62-prod-adbf9d9e4a42.herokuapp.co

End of Task 4

End of Group: Edit Data Page
Task Status: 4/4

Group

100%

Group: Delete Handling
Tasks: 3
Points: 1

COLLAPSE

Task

100%

Group: Delete Handling
Task #1: Screenshots related to delete
Weight: ~33%
Points: ~0.33

COLLAPSE

Details:
Heroku dev url must be visible in all relevant screenshots

Include ucid/date comments for each code screenshot

Columns: 1

Sub-Task

100%

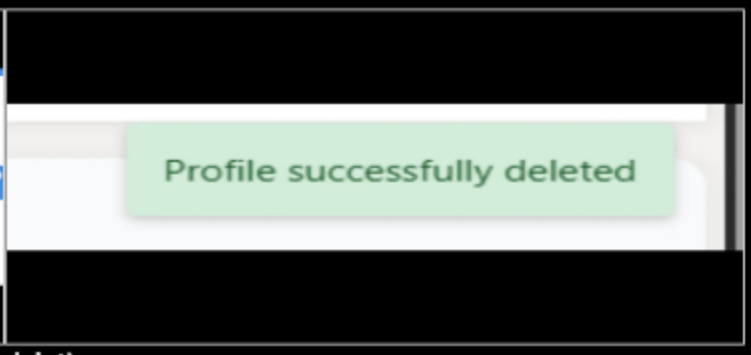
Group: Delete Handling
Task #1: Screenshots related to delete
Sub Task #1: Show the success message of a delete

Task Screenshots

Gallery Style: 2 Columns

4 2 1

confirmation



deletion

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

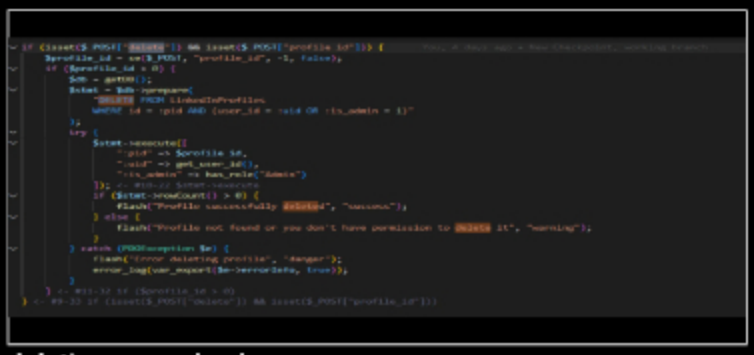
100%

Group: Delete Handling
Task #1: Screenshots related to delete
Sub Task #2: Show any error messages of a failed delete (like id not being passed)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



deletion error check

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

Checks to ensure proper rolees (admin) also makes sure that the user id exsists to be deleted and other edge case issues are at the end of the statement in else.

Sub-Task

100%

Group: Delete Handling

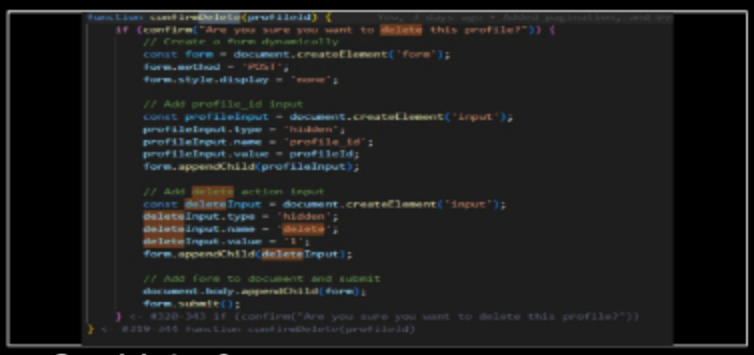
Task #1: Screenshots related to delete

Sub Task #3: Show the code related to the delete processing

Task Screenshots

Gallery Style: 2 Columns

4 2 1



confirm deletion form

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

This JavaScript function handles the deletion of a profile when called, it shows a confirmation popup asking the user if they're sure about deleting the profile. If the user confirms, the function creates a hidden HTML form dynamically in the page. This form is built with two hidden input fields: one containing the profile ID to be deleted, and another indicating this is a delete action. Once the form is constructed with these hidden fields, it's temporarily added to the webpage and automatically submitted. This making sure the deletion request is sent as a POST request rather than a GET request.

Sub-Task

100%

Group: Delete Handling
Task #1: Screenshots related to delete
Sub Task #4: Explain the delete logic

Task Response Prompt

Is it a soft or hard delete? Are there any necessary roles or restrictions? (can only delete their data, can only be done by admin, etc)?

Response:

The delete feature is a hard delete, it completely removes the row from the table on delete. The only restriction is the user needs to be logged in with someone with a role admin, and the feild being deleted needs to exists.

End of Task 1

Task

100%

Group: Delete Handling
Task #2: Screenshots of the data
Weight: ~33%
Points: ~0.33

COLLAPSE

Sub-Task

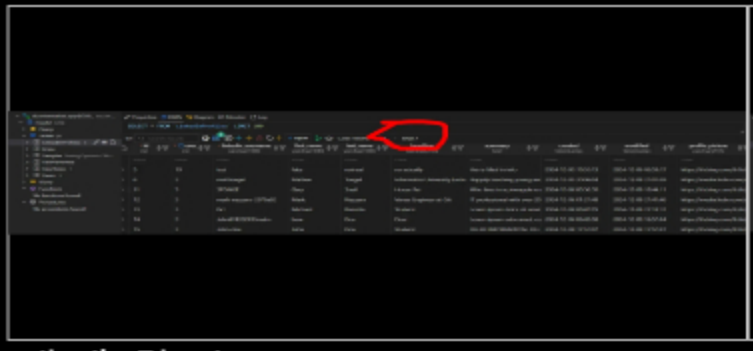
100%

Group: Delete Handling
Task #2: Screenshots of the data
Sub Task #1: Show a before and after screenshot of the DB data (two screenshots)

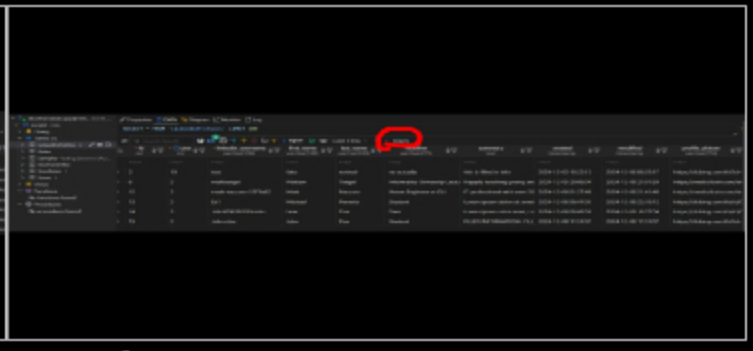
Task Screenshots

Gallery Style: 2 Columns

4 2 1



notice the 7 inputs



now see 6

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown (note precisely what changed)*

End of Task 2

Task

100%

Group: Delete Handling
Task #3: Add the pull request link for the branch related to this feature
Weight: ~33%
Points: ~0.33

COLLAPSE

Details:
Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

Task URLs

URL #1
https://github.com/Onervv/mcp62-IT202-007/pull/47

URL
https://github.com/Onervv/mcp62-IT202-007/pul

End of Task 3

End of Group: Delete Handling
Task Status: 3/3

Group

100%

Group: Misc
Tasks: 4
Points: 1

COLLAPSE

Task

100%

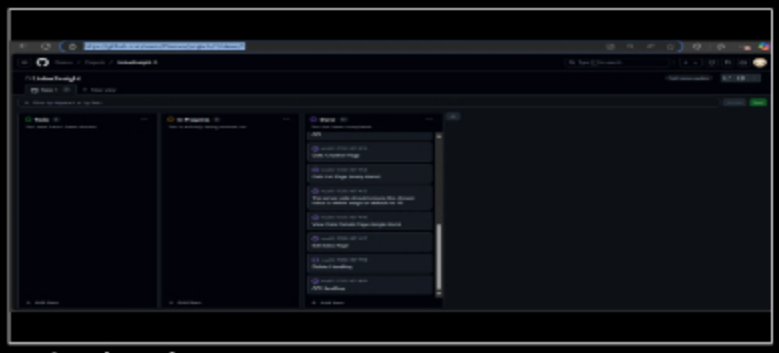
Group: Misc
Task #1: Screenshot of your project board from GitHub (tasks should be in the proper column)
Weight: ~25%
Points: ~0.25

COLLAPSE

Task Screenshots

Gallery Style: 2 Columns

4 2 1



Project board

End of Task 1

Task

100%

Group: Misc
Task #2: Provide a direct link to the project board on GitHub
Weight: ~25%
Points: ~0.25

COLLAPSE

Task URLs

URL #1
https://github.com/users/Onervv/projects/1/views/1

URL
https://github.com/users/Onervv/projects/1/view

End of Task 2

Task

100%

Group: Misc
Task #3: Talk about any issues or learnings during this assignment
Weight: ~25%
Points: ~0.25

COLLAPSE

Task Response Prompt

Response:

I was able to implement most features, but given the context of the api im accessing and the data I am fetching it didnt make sense to do certain things such as the detailed list view. But In exchnage for this I made sure the other feature I made were polished and worked well, aswell as add other things.

End of Task 3

Task

100%

Group: Misc
Task #4: WakaTime Screenshot
Weight: ~25%
Points: ~0.25

COLLAPSE

Details:

Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

Task Screenshots

Gallery Style: 2 Columns

4 2 1

wakatime

End of Task 4

End of Group: Misc
Task Status: 4/4

End of Assignment