

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-2-2024/grade/yc73>

IT202-008-S2024 - [IT202] API Project Milestone 2 2024

Submissions:

Submission Selection

1 Submission [active] 4/8/2024 2:02:04 PM

Instructions

[^ COLLAPSE ^](#)

1. Implement the Milestone 2 features from the project's proposal
2. document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>
3. Make sure you add your ucid/date as code comments where code changes are done
4. 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
5. this assignment.
6. 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.
7. Run the necessary git add, commit, and push steps to move it to GitHub
8. Complete the pull request that was opened earlier
9. Create and merge a pull request from dev to prod
Upload the same output PDF to Canvas

Branch name: Milestone2

Tasks: 29 Points: 10.00

[^ COLLAPSE ^](#)

Define Appropriate Tables for Data (1 pt.)

[^ COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of Table SQL

Checklist

*The checkboxes are for your own tracking

Points

Details

| # | Points | Details |
|----|--------|---|
| #1 | 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 | 1 | Columns should be logical and thought out (not valid to have a single field of JSON data or similar) |
| #3 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a database table with 10 columns of API data. The columns are: api_id, name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, and categoryPath. There are two additional columns at the end: stock and is_api. The table contains numerous rows of data, each representing a different item like a TV, sofa, or chair.

Table with 10 columns for the incoming API data: api_id, name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath (2 extra columns for manual inserts: stock and is_api)

Checklist Items (0)

Task #2 - Points: 1

Text: Explain the design

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Note the different fields and their purpose |

#2

1

Note if you needed to normalize the data into separate tables or if one table fit your needs

Response:

api_id = holds the unique id's for each product given by the API

name = stores the name of each product

price = holds the price of each product

measurement = holds the measurement/size of each product

typeName = stores the type of each product (it is more specific than category, e.g., dining table, 6 chairs)

image = stores the url to the main image of the product

contextualImageUrl = stores the url to the contextual image related to the product

imageAlt = holds the alternate text for the main product image

url = holds the url to the product's webpage

categoryPath = stores the category name for each product (e.g., chairs)

I was able to put this data into one table (Products) because all the fields were directly related to the products themselves and simple, meaning there was no need for multiple tables. For now, since it is not an overwhelming amount of data, a single table was enough to fit my needs.

Task #3 - Points: 1

Text: Add the pull request link for the branch related to this feature

Details:

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

URL #1

<https://github.com/yaneliii/yc73-it202-008/pull/52>

Data Creation Page (2 pts.)

[^ COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the creation page

Checklist

*The checkboxes are for your own tracking

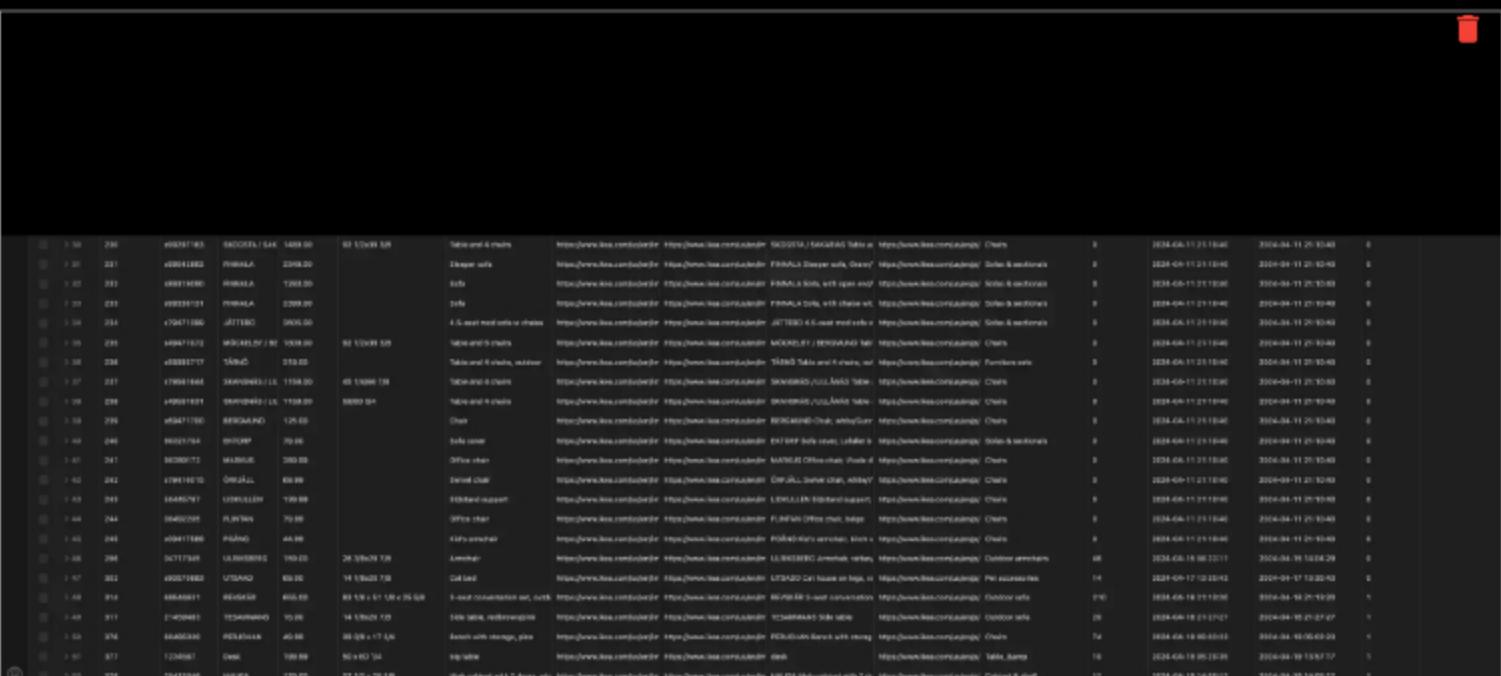
| # | Points | Details |
|----|--------|---|
| #1 | 1 | Show potentially valid data filled in for the custom creation page |
| #2 | 1 | Show how the API data is fetched for API data (must be server-side) |
| #2 | 1 | Show examples of validation messages |

| | | |
|----|---|--|
| #3 | 1 | Show an example of successful creation message |
| #4 | 1 | Design/Style should be considered (i.e., bootstrap, custom css, etc) |
| #5 | 1 | Make sure the heroku dev url is visible in the address bar |
| #6 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small
Medium
Large



X

Table before a fetch for API data

Checklist Items (1)

#1 Show potentially valid data filled in for the custom creation page

Screenshot of a browser showing a custom creation page for "Add Store Item". The page has a header with tabs for "Fetch" and "Create". The main area contains fields for "Item Name", "Description", "Category", "Price", "Stock Level", and "Image URL". Below these fields is a "Save" button.

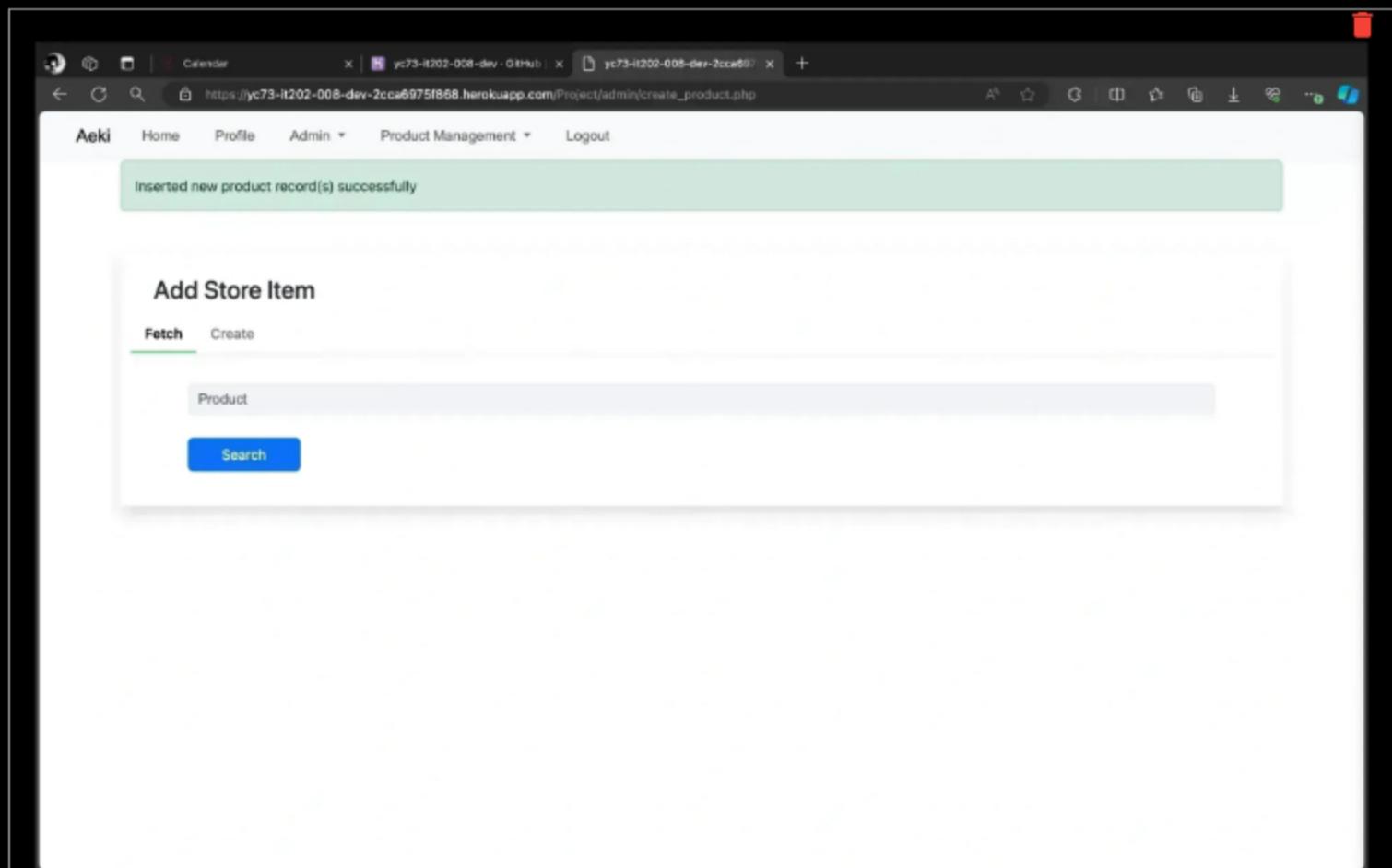
fridge

Search

Fetch form filled with valid data

Checklist Items (1)

#1 Show potentially valid data filled in for the custom creation page



Fetch form with successful creation message

Checklist Items (1)

#4 Show an example of successful creation message

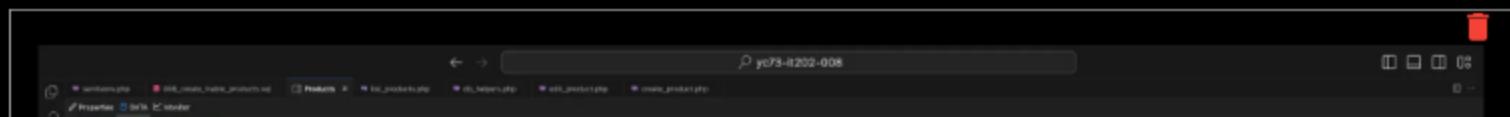


Table after successful fetch for API data

Checklist Items (1)

#4 Show an example of successful creation message

The screenshot shows a web browser window with three tabs open. The active tab is titled 'yc73-it202-008-dev · GitHub' and the URL is 'https://yc73-it202-008-dev-2cca6575f868.herokuapp.com/Project/admin/create_product.php'. The page content is as follows:

Aeki Home Profile Admin ▾ Product Management ▾ Logout

Keyword must not be empty (JS)

Add Store Item

Fetch Create

Product

Search

Fetch form: examples of validation messages

Checklist Items (1)

#3 Show examples of validation messages

Table before a manual creation

Checklist Items (1)

#1 Show potentially valid data filled in for the custom creation page

Calendar | yc73-it202-008-dev-0it- | yc73-it202-008-dev-2cc- | MATCHSPELL gaming chair- | matchspell-gaming-chair- | matchspell-gaming-chair- | +

<https://yc73-it202-008-dev-2ca6975f868.herokuapp.com/> Project/admin/create_product.php

Anti Home Profile Admin ProductManagement Logout

Add Store Item

Task: [Create](#)

Product API ID:
49027608

Product Name:
MATCHSPELL

Product Price:
289.99

Product Measurement:
21 x 62

Product Type (name):
Gaming-chair_BonitaBlack

Product Image:
https://www.lkss.in/user/images/products/matchspell-gaming-chair-bonita-black_1900x1400_1000x750_yt.jpg?h=4

Product CoverImage:
https://www.lkss.in/user/images/products/matchspell-gaming-chair-bonita-black_1900x1400_1000x750_yt.jpg?h=4

Product Image Alt:
MATCHSPELL_Gaming-chair_Bonita Black

Product URL:
<https://www.lkss.in/user/p/matchspell-gaming-chair-bonita-black-49027608>

Product Category:
Chair

Product Stock (number):

Create form filled with valid data

Checklist Items (1)

#1 Show potentially valid data filled in for the custom creation page

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "yc73-it202-008-dev-2cca6975fb86.herokuapp.com/Project/admin/create_product.php". The page displays a success message: "Inserted new product record(s) successfully". Below this, there is a form titled "Add Store Item" with a "Create" button. The form fields include "Product" and "Search".

Create form with successful creation message

Checklist Items (1)

#4 Show an example of successful creation message

The screenshot shows a database management interface with a table of data. A message at the top of the screen reads: "SELECT COUNT(*) FROM Products" and "COUNT: 346". The table has columns: ID, Name, Price, Description, Category, Status, and Created. A specific row is highlighted with a yellow background, showing the details: ID 100, Name TANMO, Price 330.00, Description Table and 6 chairs, included, Status Active, and Created 2024-04-11 21:10:40. A message at the bottom right of the table area says: "Inserted new product record(s) successfully".

Table after successful manual creation

Checklist Items (1)

#4 Show an example of successful creation message

Create form filled with incorrect data

Checklist Items (1)

#3 Show examples of validation messages

A screenshot of a web browser window showing a series of validation error messages for a product creation form. The browser has multiple tabs open, including 'matchspel-gaming-chair' and 'matchspel-gaming-chair-1'. The main content area shows a form with several input fields and their corresponding validation errors. The errors are listed in a vertical stack, each enclosed in a yellow box:

- Invalid API ID: cannot contain symbols and must be 5-20 characters (L1)
- Invalid product name: no numbers, only up to 20 characters (L2)
- Invalid price format: only up to 7 digits (L2) (e.g. 10000000 or 100000.00)
- Invalid measurement: no symbols besides / and only up to 70 characters (L2)
- Invalid product type: only up to 100 characters (L2)
- Invalid image url (L2)

The browser interface includes a header with tabs, a toolbar with various icons, and a standard address bar at the top.

Create form: examples of validation messages

Checklist Items (1)

#3 Show examples of validation messages

```
10 ?>
11
12 <?php
13     if (isset($_POST["action"])) {
14         $action = $_POST["action"];
15         $keyword = strtoupper(se($_POST, "keyword", "", false));
16         $apiID = strtoupper(se($_POST, "api_id", "", false));
17         $quote = [];
18         /* yc73 4/12/23 */
19         /* fetch logic */
20         if ($action === "fetch") {
21             if ($keyword) {
22                 $result = fetch_quote($keyword);
23                 //error_log("Data from API" . var_export($result, true));
24                 if ($result) {
25                     foreach ($result as $key => $item) {
26                         $item["is_api"] = 1;
27                         $quote[] = $item;
28                     }
29                 } <- #24-29 if ($result)
30             } <- #21-30 if ($keyword)
31         } <- #20-31 if ($action === "fetch")
32
33         /* yc73 4/12/23 */
```

Fetch: where it calls the function that processes the fetch.

Checklist Items (1)

#2 Show how the API data is fetched for API data (must be server-side)

```
1 <?php
2 // 2017-04/15/20 /?
3 function fetch_quote(Keyword) {
4     $data = ["countryCode" => "us", "keyword" => $Keyword];
5     $endpoint = "https://api-na.apis.rapidapi.com/keywordSearch";
6     $isRapidAPI = true;
7     $RapidAPIHost = "https://na.apis.rapidapi.com";
8     $result = get($endpoint, $TOKEN, $RapidAPIKey, $RapidAPIDict);
9     if ($result['status'] == 400, false == 300 && !isset($result['response'])) {
10         $result = json_decode($result['response'], true);
11     } else {
12         $result = [];
13     }
14     if (!isset($result)) {
15         $Items = $result;
16         foreach ($array_keys($Items) as $array) {
17             $key = $Items[$array];
18
19             if (isset($key['id'])) {
20                 $Items[$array]['top_id'] = $key['id'];
21                 unset($Items[$array]['id']);
22             }
23             if (is_array($key['price'])) && !isset($key['price'][currentPrice])) {
24                 $Items[$array]['price'] = $key['price'][currentPrice];
25             }
26
27             $Items[$array]['measurement'] = str_replace("", "", $key['measurement']);
28
29             $Items[$array]['imageAlt'] = str_replace("", "", $key['imageAlt']);
30
31             if (is_array($key['categoryPath'])) && !isset($key['categoryPath'][1][name])) {
32                 $Items[$array]['categoryPath'] = $key['categoryPath'][1][name];
33             }
34             if (is_array($key['variants'])) {
35                 unset($Items[$array]['variants']);
36             }
37             if (!isset($key['contextualImageAlt'])) {
38                 $Items[$array]['contextualImageAlt'] = '';
39             }
40         } <- #00--foreach ($array_keys($Items) as $array)
41         $result = $Items;
42     } <- #00-02 if (!isset($result))
43     return $result;
44 } // 4 days ago + added dot index management file(s) store, create, ~ #3-44 Function fetch_quote(Keyword)
```

Fetch: where the API data is fetched for API data (server-side)

Checklist Items (1)

#2 Show how the API data is fetched for API data (must be server-side)

Task #2 - Points: 1

Text: Screenshots of creation page code

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Form should have correct data types for each property being requested |
| #2 | 1 | Form should have correct validation for each field (HTML, JS, and PHP) |
| #3 | 1 | Successful creation should have a user-friendly message |
| #4 | 1 | Any errors should have user-friendly messages |
| #5 | 1 | Include the form/process for fetching API data |
| #6 | 1 | Include some indicator between custom data and API data |
| #7 | 1 | Include any other rules like role guards and login checks |

| | | |
|----|---|---|
| #8 | 1 | Include ucid/date comments for each code screenshot |
| #9 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

187 </div>
188 <div id="fetch" class="tab-target cr-tab-content cr-tab-fetch">
189   <!-- yc73 4/15/23 -->
190   <!-- fetch form -->
191   <form asSubmitted="return validateFetch(this)" method="POST">
192
193     <%#wp render_input([{"type": "search", "name": "Keyword", "placeholder": "Product", "rules": [{"required": true, "required": true}]}]>
194     <%#wp render_input([{"type": "hidden", "name": "action", "value": "fetch"}])>
195     <%#wp render_button([{"text": "Search", "type": "submit"}])>
196
197   </form>
198 </div>
199 <div id="create" class="tab-target cr-tab-content cr-tab-create" style="display: none;">
200   <!-- yc73 4/15/23 -->
201   <!-- create form -->
202   <form asSubmitted="return validateCreate(this)" method="POST">
203
204     <%#wp render_input([{"type": "text", "name": "api_id", "placeholder": "Product API ID", "label": "Product API ID", "rules": [{"required": true, "required": true}]}]>
205     <%#wp render_input([{"type": "text", "name": "name", "placeholder": "Product Name", "label": "Product Name", "rules": [{"required": true, "required": true}]}]>
206     <%#wp render_input([{"type": "number", "name": "price", "placeholder": "Product Price", "label": "Product Price", "rules": [{"required": true, "required": true}]}]>
207     <%#wp render_input([{"type": "text", "name": "measurement", "placeholder": "Product Measurement", "label": "Product Measurement", "rules": [{"required": true, "required": true}]}]>
208     <%#wp render_input([{"type": "text", "name": "typeCode", "placeholder": "Product type (name)", "label": "Product type (name)", "rules": [{"required": true, "required": true}]}]>
209     <%#wp render_input([{"type": "text", "name": "image", "placeholder": "Product Image", "label": "Product Image", "rules": [{"required": true, "required": true}]}]>
210     <%#wp render_input([{"type": "text", "name": "contextualImageUrl", "placeholder": "Product Contextual Image", "label": "Product Contextual Image", "rules": [{"required": true, "required": true}]}]>
211     <%#wp render_input([{"type": "text", "name": "imageAlt", "placeholder": "Product Image Alt", "label": "Product Image Alt", "rules": [{"required": true, "required": true}]}]>
212     <%#wp render_input([{"type": "url", "name": "url", "placeholder": "Product Url", "label": "Product Url", "rules": [{"required": true, "required": true}]}]>
213     <%#wp render_input([{"type": "text", "name": "categoryPath", "placeholder": "Product Category", "label": "Product Category", "rules": [{"required": true, "required": true}]}]>
214     <%#wp render_input([{"type": "number", "name": "stock", "placeholder": "Product Stock (number)", "label": "Product Stock (number)", "rules": [{"required": true, "required": true}]}]>
215
216
217     <%#wp render_input([{"type": "hidden", "name": "action", "value": "create"}])>
218     <%#wp render_button([{"text": "Search", "type": "submit", "text": "Create"}])>
219   </form>
220 </div>
221 
```

Form for Fetch and Create: showing both forms having the correct data types for each property being requested

Checklist Items (1)

#1 Form should have correct data types for each property being requested

```

183
184
185 <script>
186   /* yc73 4/15/23 */
187   /* fetch form js validation */
188   function validateFetch(form) {
189     //TODO 1: implement JavaScript validation
190     //ensure it returns false for an error and true for success
191     let is_valid = true;
192     const keyword = form.keyword.value;
193
194     if (keyword.length === 0) {
195       flash("Keyword must not be empty (JS)", "warning");
196       is_valid = false;
197     }
198   }
199 
```

```
190         is_valid = false,
191     }
192
193     return is_valid;
194 } <- #188-200 function validateFetch(form)
195
196 </script>
197
198
199
200
201
202
203
```

Fetch: showing fetch form javascript validation

Checklist Items (2)

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

#4 Any errors should have user-friendly messages



The screenshot shows a code editor with a dark theme. On the left, there's a sidebar with various icons: a file, a folder, a search bar, a refresh, a diff, a file tree, a circular progress bar, a circular progress bar with a dot, a circular progress bar with a checkmark, a database icon, a database icon with a plus, and a tree icon. The main area contains the following PHP code:

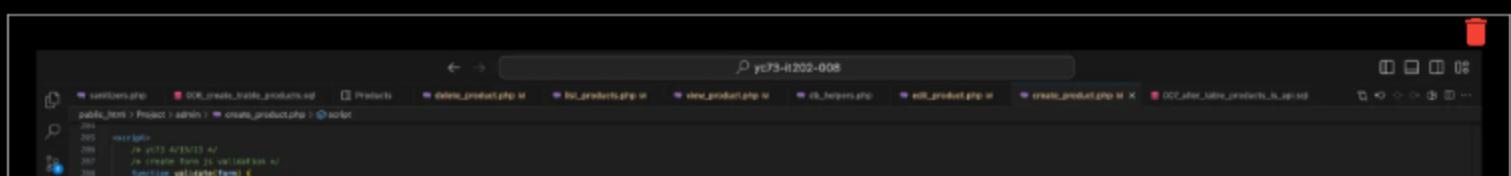
```
357
358
359     <?php
360     /* yc73 4/15/23 */
361     /* fetch form php validation */
362     //TODO 2: add PHP Code
363     if (isset($_POST["keyword"])) {
364         $keyword = se($_POST, "keyword", "", false);
365
366         //TODO 3
367         $hasError = false;
368
369         if (empty($keyword)) {
370             flash("Keyword must not be empty", "danger");
371             $hasError = true;
372         }
373     } <- #363-373 if (isset($_POST["keyword"]))
374     ?>
375
```

Fetch: showing fetch form php validation

Checklist Items (2)

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

#4 Any errors should have user-friendly messages



The screenshot shows a browser window with the URL 'yc73-h202-008'. The Network tab of the developer tools is active, showing a POST request to 'create_product.php'. The request body contains the following JSON data:

```
204
205     {
206         "id": 1234567890,
207         "name": "New Product 1234567890",
208         "description": "A new product description",
209         "price": 123.45
210     }
211
212     <-- #204-211 create_product.php validation
213     <-- #212-213 validateForm()
214     <-- #214-215 validateForm()
```

```
208 //F0010 implement Document validation
209 //assume an error_value for an error and true for success
210 let is_valid = true;
211 const REG_IS_ID = PREG_IS_ID_ID,VAL_IS_ID;
212 const REG_IS_PRIZE = PREG_IS_PRIZE_ID,VAL_IS_PRIZE;
213 const REG_IS_MEASUREMENT = PREG_IS_MEASUREMENT_ID,VAL_IS_MEASUREMENT;
214 const REG_IS_PRICE = PREG_IS_PRICE_ID,VAL_IS_PRICE;
215 const REG_IS_DATE = PREG_IS_DATE_ID,VAL_IS_DATE;
216 const REG_IS_PHONE = PREG_IS_PHONE_ID,VAL_IS_PHONE;
217 const REG_IS_EMAIL = PREG_IS_EMAIL_ID,VAL_IS_EMAIL;
218 const REG_IS_NAME = PREG_IS_NAME_ID,VAL_IS_NAME;
219 const REG_IS_CATEGORY_PATH = PREG_IS_CATEGORY_PATH_ID,VAL_IS_CATEGORY_PATH;
220 const REG_IS_STOCK = PREG_IS_STOCK_ID,VAL_IS_STOCK;
221
222 let reg_id_length_err = false;
223 let reg_id_error = "Reg ID must not be empty (S01)", "warning";
224 let id_value = false;
225
226 let reg_is_id_err = false;
227 let reg_is_id_error = "Reg ID pattern must contain symbols and must be 6-20 characters (S01)", "warning";
228 let is_id_value = false;
229
230 let reg_is_prize_err = false;
231 let reg_is_prize_error = "Reg prize no symbols and must be 6-20 characters (S01)", "warning";
232 let is_prize_value = false;
233
234 } = A228-C24 close
235
236 let name_length_err = false;
237 let name_error = "Product name must not be empty (S01)", "warning";
238 let is_name_value = false;
239
240 let reg_is_name_err = false;
241 let reg_is_name_error = "Reg product name no numbers, only up to 32 characters (S01)", "warning";
242 let is_name_value = false;
243
244 } = A239-D20 close
245
246 let price_length_err = false;
247 let price_error = "Price must not be empty (S01)", "warning";
248 let is_price_value = false;
249
250 let reg_is_price_err = false;
251 let reg_is_price_error = "Reg price pattern must be 7 digits (S01), (S01,S01),(S01,S01,S01)", "warning";
252 let is_price_value = false;
253
254 let reg_is_measurement_err = false;
255 let reg_is_measurement_error = "Reg measurement no symbols besides / and only up to 15 characters (S15)", "warning";
256 let is_measurement_value = false;
257
258 let reg_is_date_err = false;
259 let reg_is_date_error = "Reg date no symbols besides / and only up to 10 characters (S10)", "warning";
260 let is_date_value = false;
261
262 let reg_is_email_err = false;
263 let reg_is_email_error = "Reg email no symbols besides @ and only up to 25 characters (S25)", "warning";
264 let is_email_value = false;
265
266 let reg_is_phone_err = false;
267 let reg_is_phone_error = "Reg phone no symbols besides + and only up to 15 characters (S15)", "warning";
268 let is_phone_value = false;
269
270 } = A249-C28 const measurement_pattern = /
```

Create: showing create form javascript validation (1st part)

Checklist Items (2)

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

#4 Any errors should have user-friendly messages

Create: showing create form javascript validation (2nd part)

Checklist Items (2)

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

#4 Any errors should have user-friendly messages

The screenshot shows a code editor with several tabs open, including `editors.php`, `004_create_product_products.php`, `Products`, `delete_product.php`, `list_products.php`, `view_product.php`, `db_helpers.php`, `edit_product.php`, and `create_product.php`. The current file is `create_product.php`. The code contains validation logic for various fields:

```
300 if (alt.length === 0) {
301   flash("Image alt text must not be empty (35)", "warning");
302   is_valid = false;
303 }
304
305 // yc73 4/15/23 nf
306 // create form js validation nf
307 if (url.length === 0) {
308   flash("Product url must not be empty (35)", "warning");
309   is_valid = false;
310 }
311 else {
312   const url_pattern = /(https?://(www\.)?[-a-zA-Z-9@%_.+~#=]{2,256})[-a-zA-Z-9@%_.+~#=]{2,6}\.([-a-zA-Z-9@%_.+~#=]{1,63})\.([a-zA-Z-9@%_.+~#=]{1,3})/i;
313   if(url_pattern.test(url)) {
314     flash("invalid product url (35)", "warning");
315     is_valid = false;
316   }
317 } ← #022-327 [-a-zA-Z-9@%_.+~#=]{1,63}([-a-zA-Z-9@%_.+~#=]{1,3})/i;
318
319 if (category.length === 0) {
320   flash("Category must not be empty", "warning");
321   is_valid = false;
322 }
323 else {
324   const category_pattern = /^[a-zA-Z]{1,100}$/;
325   if(category_pattern.test(category)) {
326     flash("invalid Category path no symbols besides _ and &, only up to 100 characters (35)", "warning");
327     is_valid = false;
328   }
329 } ← #033-328 else;
330
331 if (stock.length === 0) {
332   flash("Stock must not be empty (35)", "warning");
333   is_valid = false;
334 }
335 else {
336   const stock_pattern = /^[0-9]+$/;
337   if(stock_pattern.test(stock)) {
338     flash("invalid stock! only numbers (35)", "warning");
339     is_valid = false;
340   }
341 } ← #045-350 else;
342
343 return is_valid;
344 } ← #002-354 const url_pattern = /(https?://(www\.)?[-a-zA-Z-9@%_.+~#=]{2,256})[-a-zA-Z-9@%_.+~#=]{2,6}\.([-a-zA-Z-9@%_.+~#=]{1,63})\.([a-zA-Z-9@%_.+~#=]{1,3})/i;
345
346 </script>
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
845
846
847
847
848
849
849
850
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
862
863
863
864
865
865
866
867
867
868
868
869
869
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1650
1651
1651
1652
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1660
1661
1661
1662
1662
1663
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1670
1671
1671
1672
1672
1673
1673
1674
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1680
1681
1681
1682
1682
1683
1683
1684
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1690
1691
1691
1692
1692
1693
1693
1694
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1700
1701
1701
1702
1702
1703
1703
1704
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1710
1711
1711
1712
1712
1713
1713
1714
1714
1715
1715
1716
1716
1717
1717
1718
1718
1
```

```

487     flash["Invalid product name: no numbers, only up to 32 characters", "danger"];
488     $hasError = true;
489   }
490
491   if (empty($price)) {
492     flash["Price must not be empty", "danger"];
493     $hasError = true;
494   }
495
496   if (!is_valid_price($price)) {
497     flash["Invalid price format: only up to 7 digits (ex: XXXXXX or XXXXX.XX)", "danger"];
498     $hasError = true;
499   }
500
501   if (empty($measurement)) {
502     flash["Measurement must not be empty", "danger"];
503     $hasError = true;
504   }
505
506   if (!is_valid_measurement($measurement)) {
507     flash["Invalid measurement: no symbols besides / and only up to 35 characters", "danger"];
508     $hasError = true;
509   }
510
511   if (empty($image)) {
512     flash["Image url must not be empty", "danger"];
513     $hasError = true;
514   }
515
516   if (!is_valid_image_url($image)) {
517     flash["Invalid image url", "danger"];
518     $hasError = true;
519   }
520
521   if (empty($link)) {
522     flash["Link url must not be empty", "danger"];
523     $hasError = true;
524   }
525
526   if (!is_valid_link($link)) {
527     flash["Invalid link url", "danger"];
528     $hasError = true;
529   }
530
531   if (empty($category)) {
532     flash["Category must not be empty", "danger"];
533     $hasError = true;
534   }
535
536   if (!is_valid_category($category)) {
537     flash["Invalid category path: no symbols besides _ and & only up to 100 characters", "danger"];
538     $hasError = true;
539   }
540
541   if (empty($stock)) {
542     flash["Stock must not be empty", "danger"];
543     $hasError = true;
544   }
545
546   if (!is_valid_stock($stock)) {
547     flash["Invalid stock: only numbers", "danger"];
548     $hasError = true;
549   }
550
551 } --> $hasError = false;
552
553 return $hasError;
554
555 function is_valid_link($link) {
556   let target = document.createElement('a');
557   let href = target.href;
558
559   if (target.href === href) {
560     return true;
561   }
562
563   return false;
564 }

```

Create: showing create form php validation (1st part)

Checklist Items (2)

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

#4 Any errors should have user-friendly messages

```

487     flash["Invalid product name: no numbers, only up to 32 characters", "danger"];
488     $hasError = true;
489   }
490
491   if (empty($price)) {
492     flash["Price must not be empty", "danger"];
493     $hasError = true;
494   }
495
496   if (!is_valid_price($price)) {
497     flash["Invalid price format: only up to 7 digits (ex: XXXXXX or XXXXX.XX)", "danger"];
498     $hasError = true;
499   }
500
501   if (empty($measurement)) {
502     flash["Measurement must not be empty", "danger"];
503     $hasError = true;
504   }
505
506   if (!is_valid_measurement($measurement)) {
507     flash["Invalid measurement: no symbols besides / and only up to 35 characters", "danger"];
508     $hasError = true;
509   }
510
511   if (empty($image)) {
512     flash["Image url must not be empty", "danger"];
513     $hasError = true;
514   }
515
516   if (!is_valid_image_url($image)) {
517     flash["Invalid image url", "danger"];
518     $hasError = true;
519   }
520
521   if (empty($link)) {
522     flash["Link url must not be empty", "danger"];
523     $hasError = true;
524   }
525
526   if (!is_valid_link($link)) {
527     flash["Invalid link url", "danger"];
528     $hasError = true;
529   }
530
531   if (empty($category)) {
532     flash["Category must not be empty", "danger"];
533     $hasError = true;
534   }
535
536   if (!is_valid_category($category)) {
537     flash["Invalid category path: no symbols besides _ and & only up to 100 characters", "danger"];
538     $hasError = true;
539   }
540
541   if (empty($stock)) {
542     flash["Stock must not be empty", "danger"];
543     $hasError = true;
544   }
545
546   if (!is_valid_stock($stock)) {
547     flash["Invalid stock: only numbers", "danger"];
548     $hasError = true;
549   }
550
551 } --> $hasError = false;
552
553 return $hasError;
554
555 function is_valid_link($link) {
556   let target = document.createElement('a');
557   let href = target.href;
558
559   if (target.href === href) {
560     return true;
561   }
562
563   return false;
564 }

```

Create: showing create form php validation (2nd part)

Checklist Items (2)

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

#4 Any errors should have user-friendly messages

```

104  */
105
106 /* yc73 4/12/23 */
107 /* insert data logic */
try {
108
109   //optional options for debugging and duplicate handling

```

```

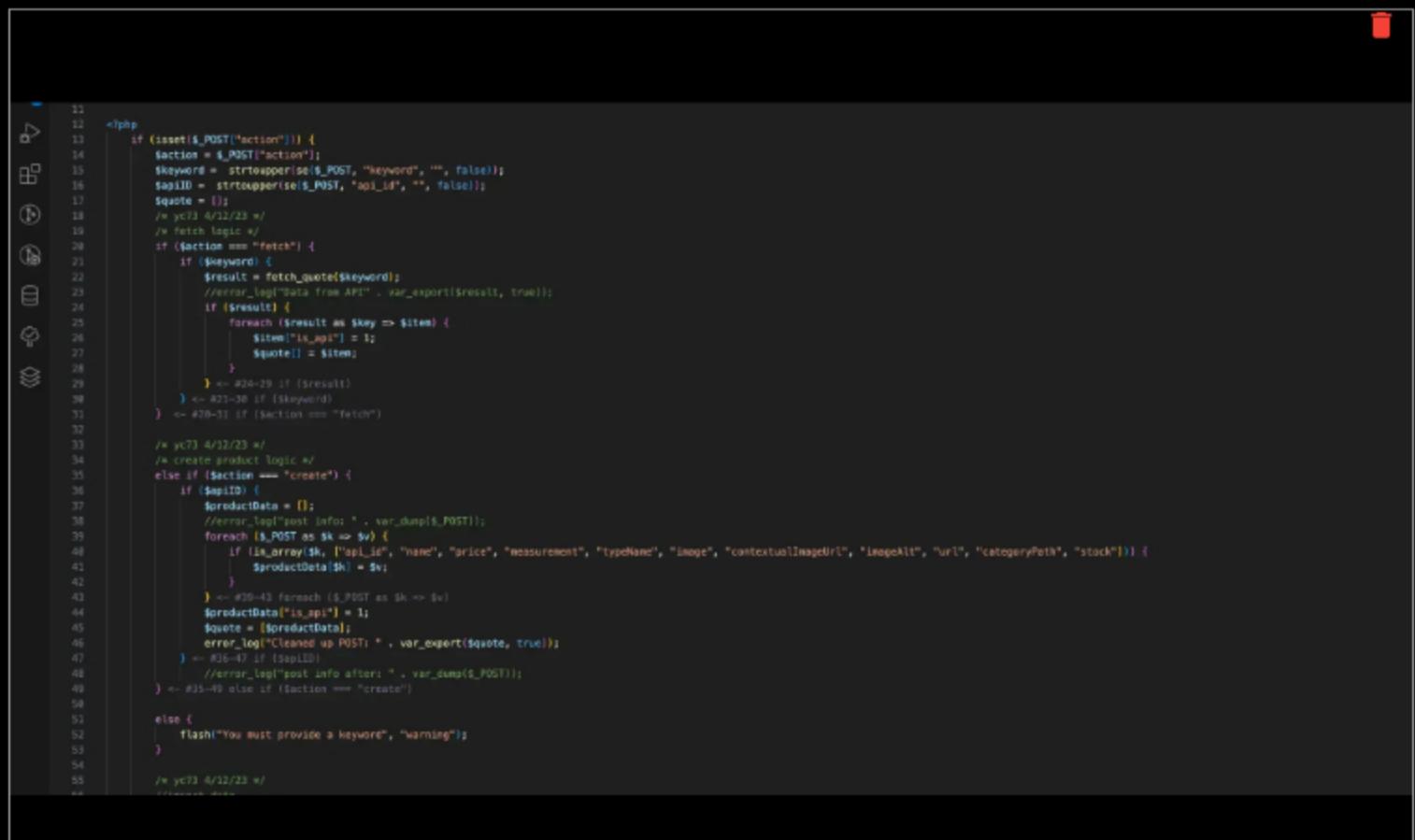
100 // optional options for debugging and duplicate handling
101
102 $opts =
103     ["debug" => true, "update_duplicate" => false, "columns_to_update" => []];
104
105 $result = insert("Products", $quote, $opts);
106 if (!$result) {
107     flash("Unhandled error", "warning");
108 } else {
109     //flash("Created record with id " . var_export($result, true), "success");
110     flash("Inserted new product record(s) successfully", "success");
111 }
112
113 } catch (InvalidArgumentException $e1) {
114     error_log("Invalid arg" . var_export($e1, true));
115     flash("Invalid data passed", "danger");
116 } catch (PDOException $e2) {
117     if ($e2->errorInfo[1] == 1062) {
118         flash("An entry for this product already exists", "warning");
119     } else {
120         error_log("Database error" . var_export($e2, true));
121         flash("Database error", "danger");
122     }
123
124 } catch (Exception $e3) {
125     error_log("Invalid data records" . var_export($e3, true));
126     flash("Invalid data records", "danger");
127 }
128
129 } <- #13-133 if (isset($_POST["action"]))
130
131
132
133 } <- #13-133 if (isset($_POST["action"]))
134
135

```

Fetch and Create: Showing the insert logic and (user-friendly) successful creation message

Checklist Items (1)

#3 Successful creation should have a user-friendly message



```

11 <?php
12 if (isset($_POST["action"])) {
13     $action = $_POST["action"];
14     $keyword = strtoupper($_POST["keyword"], "", false);
15     $apiID = strtoupper($_POST["api_id"], "", false);
16     $quote = [];
17
18     /* yc73 4/12/23 */
19
20     /* fetch logic */
21     if ($action === "fetch") {
22         if ($keyword) {
23             $result = fetch_quoted($keyword);
24             //error_log("Data from API" . var_export($result, true));
25             if ($result) {
26                 foreach ($result as $key => $item) {
27                     $item["is_api"] = 1;
28                     $quote[] = $item;
29                 }
30             } <- #26-29 if ($result)
31         } <- #21-30 if ($keyword)
32     } <- #20-31 if ($action === "fetch")
33
34     /* yc73 4/12/23 */
35     /* create product logic */
36     else if ($action === "create") {
37         if ($apiID) {
38             $productData = [];
39             //error_log("Post info: " . var_dump($_POST));
40             foreach ($_POST as $k => $v) {
41                 if (!in_array($k, ['api_id', 'name', 'price', 'measurement', 'typeName', 'image', 'contextualImageUrl', 'imageAlt', 'url', 'categoryPath', 'stack'])) {
42                     $productData[$k] = $v;
43                 }
44             } <- #39-43 foreach ($_POST as $k => $v)
45             $productData["is_api"] = 1;
46             $quote = [$productData];
47             error_log("Cleaned up POST: " . var_export($quote, true));
48         } <- #36-47 if ($apiID)
49             //error_log("Post info after: " . var_dump($_POST));
50     } <- #35-49 else if ($action === "create")
51
52     else {
53         flash("You must provide a keyword", "warning");
54     }
55
56     /* yc73 4/12/23 */
57
58

```

Showing the logic for fetch and create

Checklist Items (1)

"#3 Fetch and Create logic for creating API data"

#5 Include the form/process for fetching API data

The screenshot shows a code editor with multiple tabs open. The active tab is `store_api.php`, which contains the following PHP code:

```
1 // store_api.php > fetch_quote
2
3 function fetch_quote($keyword) {
4     $data = ["countryCode" => "us", "keyword" => $keyword];
5     $endpoint = "https://ikea-api.p.rapidapi.com/keywordSearch";
6     $useRapidAPI = true;
7     $rapidAPIKey = "ikea-api.p.rapidapi.com";
8     $result = get($endpoint, "STORE_API_KEY", $data, $useRapidAPI, $rapidAPIKey);
9     if ($result["status"] == 200 && !isset($result["response"])) {
10         $result = json_decode($result["response"], true);
11     } else {
12         $result = [];
13     }
14     if (!isset($result)) {
15         $items = $result;
16         foreach ($array_keys($items) as $array) {
17             $key = $items[$array];
18
19             if (isset($key['id'])) {
20                 $items[$array]['api_id'] = $key['id'];
21                 unset($items[$array]['id']);
22             }
23             if (!is_array($key['price']) && !isset($key['price'])) {
24                 $items[$array]['price'] = $key['price'];
25             }
26
27             $items[$array]['measurement'] = str_replace("", "", $key['measurement']);
28
29             $items[$array]['imageAlt'] = str_replace("", "", $key['imageAlt']);
30
31             if (!is_array($key['categoryPath']) && !isset($key['categoryPath'][0])) {
32                 $items[$array]['categoryPath'] = $key['categoryPath'][0];
33             }
34             if (!is_array($key['variants'])) {
35                 unset($items[$array]['variants']);
36             }
37             if (!isset($key['contextualImageUrl'])) {
38                 $items[$array]['contextualImageUrl'] = "";
39             }
40         }
41     } // #30-40 foreach ($array_keys($items) as $array)
42     $result = $items;
43 } // #41-42 if (!isset($result))
44 return $result;
45 }
```

The code is annotated with comments explaining the process of fetching data from the IKEA API and transforming it into a structured array.

Showing function that does the process for fetching API data

Checklist Items (1)

#5 Include the form/process for fetching API data

The screenshot shows a database browser interface with the `Products` table selected. The table has the following columns:

| image | contextualImageUrl | imageAlt | url | categoryPath | stock | createdTimestamp | modifiedTimestamp | isApi |
|-------------------------------|-------------------------------|---------------------------------|-------------------------------|--------------------|-------|---------------------|---------------------|------------|
| text | text | text | text | varchar(100) | int | timestamp | timestamp | tinyint(1) |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | TÄRNÖ Table and 4 chairs, oak | https://www.ikea.com/us/en/p/ | Furniture sets | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | SKANSNÄS / LILLÅNÄS Table | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | SKANSNÄS / LILLÅNÄS Table | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | BERGMUND Chair, white/Gunn | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | EXTORP Sofa cover, Loftslit b | https://www.ikea.com/us/en/p/ | Sofas & sectionals | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | MARKUS Office chair, Vissle d | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | ÖRFJÄLL Swivel chair, white/V | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | LIDKÜLLEN St/stand support, | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | FLINTAN Office chair, beige | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | PÖÄNG Kid's armchair, birch v | https://www.ikea.com/us/en/p/ | Chairs | 0 | 2024-04-11 21:10:40 | 2024-04-11 21:10:40 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | ULRIKSBERG Armchair, rattan | https://www.ikea.com/us/en/p/ | Outdoor armchairs | 46 | 2024-04-15 08:22:11 | 2024-04-15 14:04:29 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | UTSADD Cat house on legs, II | https://www.ikea.com/us/en/p/ | Pet accessories | 14 | 2024-04-17 15:35:43 | 2024-04-17 15:35:43 | 0 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | REVKÄR 3-seat conversation | https://www.ikea.com/us/en/p/ | Outdoor sofa | 210 | 2024-04-18 21:18:20 | 2024-04-18 21:18:20 | 1 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | TESAMMANS Side table | https://www.ikea.com/us/en/p/ | Outdoor sofa | 20 | 2024-04-18 21:27:27 | 2024-04-18 21:27:27 | 1 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | PERJOHAN Bench with storage | https://www.ikea.com/us/en/p/ | Chairs | 74 | 2024-04-19 05:02:23 | 2024-04-19 05:02:23 | 1 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | desk | https://www.ikea.com/us/en/p/ | Table_& | 10 | 2024-04-19 05:20:35 | 2024-04-19 19:57:17 | 1 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | HAUGA High cabinet with 2 doors | https://www.ikea.com/us/en/p/ | Cabinet & shelf | 12 | 2024-04-19 14:06:12 | 2024-04-19 14:06:12 | 1 |
| www.ikea.com/us/en/images/p | www.ikea.com/us/en/images/p | table | www.ikea.com/us/en/p/ | Side & lamp | 26 | 2024-04-19 16:23:02 | 2024-04-19 16:38:58 | 1 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | STJÄRNSTATUS French door r | https://www.ikea.com/us/en/p/ | Appliances | 0 | 2024-04-19 18:35:00 | 2024-04-19 18:35:00 | 1 |
| https://www.ikea.com/us/en/in | https://www.ikea.com/us/en/in | FÄRSKJET Bottom-freezer ref | https://www.ikea.com/us/en/p/ | Appliances | 0 | 2024-04-19 18:35:00 | 2024-04-19 18:35:00 | 1 |



Showing API indicator in the table: the column is_api (0 for yes, 1 for no)

Checklist Items (1)

#6 Include some indicator between custom data and API data

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for db_helpers.php, edit_product.php, create_product.php, store_api.php, and 007_alter_table_products_is_api.sql. The current tab is 007_alter_table_products_is_api.sql. The SQL editor contains the following code:

```
1 /* yc73 4/15/23 */
2 ALTER TABLE `Products`
3 ADD COLUMN `is_api` tinyint(1) NOT NULL DEFAULT 0;
```

The status bar at the bottom shows the file is007_alter_table_products_is_api.sql, the connection is202, and the user yc73.

Showing API indicator column: added the column is_api

Checklist Items (1)

#6 Include some indicator between custom data and API data

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 16_create_table_products.sql, Products, list_products.php, db_helpers.php, edit_product.php, create_product.php, and 007_alter_table_products_is_api.sql. The current tab is 007_alter_table_products_is_api.sql. The SQL editor contains the following code:

```
1 /* yc73 4/1/23 */
2 if (!has_role("Admin")) {
3     flash("You don't have permission to view this page", "warning");
4     die(header("Location: $BASE_PATH" . "/home.php"));
5 }
```

The status bar at the bottom shows the file is007_alter_table_products_is_api.sql, the connection is202, and the user yc73.

```

1.0
11
12 <?php

```

Showing a role guard for the creation page

Checklist Items (1)

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

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Show at least one record fetched from the API |
| #2 | 1 | Show at least one record created via the creation form |
| #3 | 1 | Note what differs |
| #4 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a database browser interface with a table named 'products'. The table has columns: id, name, price, description, and category_id. There are approximately 20 rows of data, each containing a unique product ID, name, price, description, and a category ID. The interface includes a toolbar with various icons for database management, a properties panel on the left, and a search bar at the top.

Fetch: showing records fetched from the API Difference: it inserted 12 product records into the table (it also has a 1 for the is_api column - as my choice to indicate it's not api since I manually fetched it) (the ones highlighted in blue)

Checklist Items (1)

#1 Show at least one record fetched from the API

Create: showing one record created via the creation form Fetch: showing records fetched from the API Difference: it inserted 1 product record into the table (it also has a 1 for the is_api column - as my choice to indicate it's not api since I manually created it) (the insert is all the way at the bottom in blue)

Checklist Items (1)

#2 Show at least one record created via the creation form

Task #1 - Points: 1

Text: Explain the process

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Provide a high-level step-by-step of how fetching API data works and gets added to your DB (include how duplicates are handled) |
| #2 | 1 | Provide a high-level step-by-step of how creating custom data works and gets added to your DB (include how duplicates are handled) |
| #3 | 1 | Briefly describe the validations for the applicable fields |
| #4 | 1 | Describe how duplicate data is handled |

Response:

To fetch, a user must enter a (product) keyword. When they submit that fetch form, it will trigger the "onsubmit" event and call the function "validateFetch". This function is the client-side javascript validation which checks the (product) keyword field.

- First, it grabs the value of what the user entered into the field (named) keyword and stores the value into a corresponding variable (keyword).
- It proceeds to check if the field is empty by looking at the length of the variable with ".length==0".

The "onsubmit" takes a boolean value. So, if it fails this validation it returns a false, then displays a warning message to the user, and the form is not submitted. However, if it does NOT fail this validation, it returns true and sends the data with the POST method.

So, if the client-side validation is valid (true), the form data is sent to the server-side, and it goes through PHP validations.

- First, it checks if the POST array contains the keyword. If it does, it retrieves the data, then sets it to a variable and begins validating the data.
- It proceeds to check if the keyword is empty by using an empty() function.

If the field fails the validation, it returns true for "hasError" and flashes error messages to the user. However, if the field passes the validation, it returns false for "hasError" (meaning there are no errors) and proceeds. (p.s. Since I cannot see all the keywords that the API uses I wasn't sure what other restrictions I should put. So, in fear of breaking it, I left the js and php validations at this.)

So, along with the keyword, a hidden input called "action" is sent with the value "fetch". This "action" is checked by an if-statement to see if it was set. If it is, then it proceeds to grab the value of the keyword (entered by the user). Another if-statement checks if "action" contains the value "fetch" to run the code for fetch.

The user's keyword is passed to the function "fetch_quote" - which actually processes the fetch from the API.

- "fetch_quote" puts the required API parameters (including the keyword), API endpoint, and API host into separate variables so it's easier to pass.
- Then it calls get () and sends a request to the API endpoint along with my API key, required API parameters, and the API host. It locks this code until it gets a response or times out.

If we get a successful response with data, it decodes (JSON) the response string and converts it to an associative array stored in the variable "result".

Next, it checks if the "result" is set. If it IS it can perform some data transformations. It loops over each array to: rename the "id" to "api_id" (removing "id" after setting the value to "api_id"), extract the value of 'currentPrice' and set it as the value of "price", remove the double quotes from "measurement" and "imageAlt", extract the category 'name' and

as the value of "price", remove the double quotes from "measurement" and "imageAlt", extract the category name and set it as the value for "categoryPath", remove the key "variants" entirely, and setting an empty string for 'contextualImageUrl' if it doesn't exist in that array.

Then, it can finally return the processed result.

Another if-statement checks "result" contains data after being returned. If it does, then a foreach loop iterates over each item in "result" creating key-value pairs (to match the DB table), and puts them into an array "quote".

The transformed data "quote" is passed to a function "insert" which handles inserting the API data.

- First, it validates the array information, makes sure the array is not empty, makes sure it has the proper table name, and makes sure the table is a string.
- Next, it determines if the array is indexed with associative array records, meaning it is a multi-insert, and checks the data structure before proceeding so nested arrays aren't indexed.
- Then, it starts to build a query by sorting the keys (in the array) to match the column order to ensure consistency.
- It dynamically puts that query together and attempts to create an INSERT INTO query based on the table, columns, and what's being passed in.
- Also to handle duplicate keys, it will update any existing records with the new data by using an ON DUPLICATE KEY UPDATE process.
- Then, it proceeds to prepare the query and bind all the data (with bindValue). It is going to try and execute the query, return the number of rows affected and the last insert ID. If it is successful, it will flash a user-friendly success message to the user.

To create custom data, a user must enter an api_id, name, price, measurement, type, image, contextual image, alt text, url category, and stock number. When they submit that create form, it will trigger the "onsubmit" event and call the function "validateFetch". This function is the client-side javascript validation which checks the (product) keyword field.

- First, it grabs the values of what the user entered into each field and stores the value into a corresponding variable (api_id, name, price, measurement, typeName, img, contextualImageUrl, alt, url, category, stock).
- It proceeds to check if the fields are empty, it looks at the length of the variable with ".length==0".
- Using regex: For api_id, it also checks if it contains symbols, if shorter than 6 or longer than 20 characters. For name, it also checks if it contains numbers or is longer than 32 characters. For price, it also checks if there's anything other than numbers and if it's longer than 7 digits. For measurement, it checks if it contains symbols or is longer than 75 characters. For type, it checks if it's longer than 100 characters. For image, contextual image, and url, it checks if the format of the URLs is valid. For category, it also checks if it contains symbols (other than _ and &) or if it's longer than 100 characters. For stock, it checks if there's anything other than numbers.

The "onsubmit" takes a boolean value, so if any of these validations returns a false, then warning messages are displayed to the user, and the form is not submitted. However, if none of the validations fail, it returns true and sends the data with the POST method.

So, if the client-side validation is valid (true), the form data is sent to the server-side, and it goes through PHP validations.

- First, it checks if the POST array contains api_id, name, price, measurement, typeName, img, contextualImageUrl, alt, url, category, stock. If it does, it retrieves the data, then sets it to variables and begins validating the data.
 - It proceeds to check if any of the variables are empty by using an empty() function.
- Also using the same regex as in the js validations: For api_id, it also checks if it contains symbols, if shorter than 6 or longer than 20 characters. For name, it also checks if it contains numbers or is longer than 32 characters. For price, it also checks if there's anything other than numbers and if it's longer than 7 digits. For measurement, it checks if it contains symbols or is longer than 75 characters. For type, it checks if it's longer than 100 characters. For image, contextual image, and url, it checks if the format of the URLs is valid. For category, it also checks if it contains symbols (other than _ and &) or if it's longer than 100 characters. For stock, it checks if there's anything other than numbers.

than 6 or longer than 20 characters. For name, it also checks if it contains numbers or is longer than 32 characters. For price, it also checks if there's anything other than numbers and if it's longer than 7 digits. For measurement, it checks if it contains symbols or is longer than 75 characters. For type, it checks if it's longer than 100 characters. For image, contextual image, and url, it checks if the format of the URLs is valid. For category, it also checks if it contains symbols (other than _ and &) or if it's longer than 100 characters. For stock, it checks if there's anything other than numbers.

If the field fails the validation, it returns true for "hasError" and flashes error messages to the user. However, if the field passes the validation, it returns false for "hasError" (meaning there are no errors) and proceeds.

So, along with the keyword, a hidden input called "action" is sent with the value "create". This "action" is checked by an if-statement to see if it was set. If it is, then it proceeds to grab the value of the api_id (entered by the user). Another if-statement checks if "action" contains the value "create" to run the code for creating custom data.

If the field "api_id" is not empty, it proceeds to make an empty array stored in the variable "productData" - which is later used to hold the fixed-up form data.

- A foreach loop iterates over all of the (post) data submitted by the user and assigns the corresponding value to the same key - all put together into "productData" (is_api is also added with the value of 1 to indicate it is manually created). Then, "productData" is put into another array "quote".

The "quote" is passed to a function "insert" which handles inserting the data into the table.

- First, it validates the array information, makes sure the array is not empty, makes sure it has the proper table name, and makes sure the table is a string.
- Next, it checks the data structure and determines if the array is a single associative array, meaning it is a single insert, and checks if the values are not arrays/objects before proceeding.
- Then, it starts to build a query by sorting the keys (in the array) to match the column order to ensure consistency.
- It dynamically puts that query together and attempts to create an INSERT INTO query based on the table, columns, and what's being passed in.
- Also to handle duplicate entries, if it catches a 1062 db error, meaning it is a duplicate entry, it will flash the user a warning message saying that the product already exists.
- If no errors occur, then it proceeds to prepare the query and bind all the data (with bindValue). It is going to try and execute the query, return the number of rows affected and the last inter ID. If it is successful, it will flash a user-friendly success message to the user.

Task #5 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Include the heroku prod link for this page |
| #2 | 1 | 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 |

URL #1

<https://www.heroku.com/deploy> "MFG"

<https://github.com/yaneliii/yc73-it202-008/pull/53>

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/56>

URL #3

<https://github.com/yaneliii/yc73-it202-008/pull/57>

URL #4

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/create_product.php

Data List Page (many entities) (2 pts.)

[^ COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the list page

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Show the page of your entities listed (have a reasonable number shown) |
| #2 | 1 | Show the filter/sort form based on your data and the required limit field |
| #3 | 1 | Demonstrate a few varied filters/sorts |
| #4 | 1 | Demonstrate a filter that doesn't have any records (should show an appropriate message) |
| #5 | 1 | 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) |
| #6 | 1 | Each list item should have a summary of the entity (likely won't be the entire entity data) |
| #7 | 1 | Design/Style should be considered (i.e., bootstrap, custom css, etc) |
| #8 | 1 | Make sure the heroku dev url is visible in the address bar |
| #9 | 1 | Clearly caption screenshots |

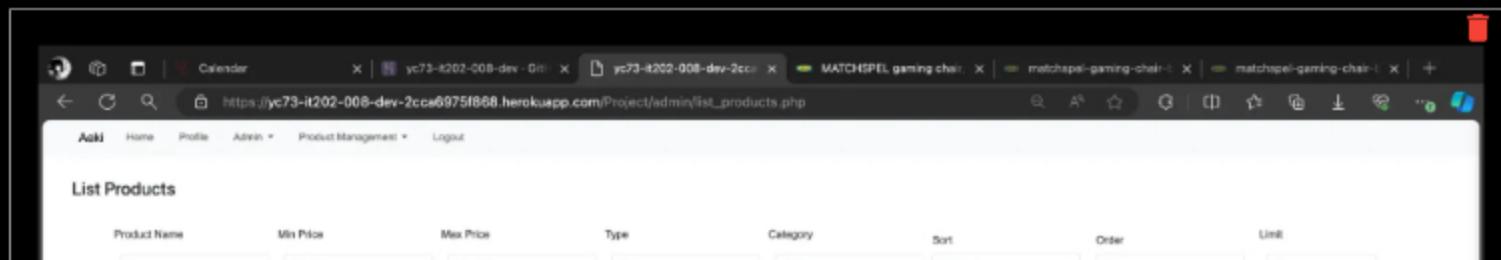
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



| Name | Min Price | Max Price | Type | Category | Created | | |
|----------|-----------|-----------|--------------------------------|-------------------------------|---------|---|---|
| KALLAX | 79.99 | | Shelf unit | Living room & entryway tables | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| Filter | | | | | | | |
| LACK | 19.99 | | TV unit | TV & media furniture | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FRIHETEN | 699.00 | | Sleeper sectional,3 seat sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| UPPLAND | 449.00 | | Armchair | Chairs | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LIMMÅS | 300.00 | | Sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| JÖKKMOKK | 199.99 | | Table and 4 chairs | Chairs | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| KIVIK | 1449.00 | | Sectional, 5-seat corner | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FRIHETEN | 749.00 | | Sleeper sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FIMNÄLA | 1699.00 | | Sectional, 4-seat | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| UPPLAND | 840.00 | | Sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| BNÄS | 183.99 | | TV unit | TV & media furniture | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| KIVIK | 549.00 | | 3-seat sleeper sofa | Chairs | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| GLOSTAD | 149.00 | | Loveseat | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| TROFAST | 79.99 | | Storage combination with boxes | Nursery furniture | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LIMMÅS | 230.00 | | Sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| MÄLAR | 340.00 | | High bed frame/2 storage boxes | Beds | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| POÄNG | 99.00 | | Armchair | Chairs | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| HOLMSUND | 1149.00 | | Sleeper sectional, 3-seat | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> |

Showing page with multiple entities listed and the filter/sort

Checklist Items (1)

#1 Show the page of your entities listed (have a reasonable number shown)

| List Products | | | | | | | |
|---------------|-----------|------------------------------------|--------------------|----------|---------|---|-------|
| Product Name | Min Price | Max Price | Type | Category | Sort | Order | Limit |
| FIMNÄLA | Min Price | Max Price | Type | Category | Created | + | 50 |
| Filter | | | | | | | |
| name | price | typeName | categoryPath | stock | is_api | Actions | |
| FIMNÄLA | 144.00 | Cover for loveseat/sleeper section | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 2399.00 | Sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 1289.00 | Sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 2349.00 | Sleeper sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 1686.00 | Sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 1946.00 | 3-seat sleeper sofa with chaise | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 2749.00 | Sleeper sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 1949.00 | Sleeper sofa | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 1048.00 | Loveseat | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |
| FIMNÄLA | 1699.00 | Sectional, 4-seat | Sofas & sectionals | 0 | 0 | <button>View</button> <button>Edit</button> <button>Delete</button> | |

Showing filter = Name, sort = Created, order = ascending +, limit = 10

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

Aekli Home Profile Admin Product Management Logout

List Products

| Product Name | Min Price | Max Price | Type | Category | Sort | Order | Limit |
|--------------|-----------|-----------|------|----------|---------|-------|-------|
| finnala | 1000 | 1500 | sofa | Category | Created | + | 10 |

Filter **Clear**

| name | price | typeName | categoryPath | stock | is_api | Actions |
|---------|---------|--------------|--------------------|-------|--------|--|
| FINNALA | 1099.00 | Sleeper sofa | Sofas & sectionals | 0 | 0 | View Edit Delete |
| FINNALA | 1142.00 | Sofa | Sofas & sectionals | 0 | 0 | View Edit Delete |
| FINNALA | 1184.00 | Sofa | Sofas & sectionals | 0 | 0 | View Edit Delete |
| FINNALA | 1263.00 | Sofa | Sofas & sectionals | 0 | 0 | View Edit Delete |

Showing: filter = Name, Min Price, Max Price, Type sort = Created, order = ascending +, limit = 10

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

Aekli Home Profile Admin Product Management Logout

List Products

| Product Name | Min Price | Max Price | Type | Category | Sort | Order | Limit |
|--------------|-----------|-----------|------|----------|---------|-------|-------|
| malm | 60 | 200 | Type | desks | Created | + | 10 |

Filter **Clear**

| name | price | typeName | categoryPath | stock | is_api | Actions |
|------|--------|--------------------------|----------------|-------|--------|--|
| MALM | 129.99 | Dressing table | Tables & desks | 0 | 0 | View Edit Delete |
| MALM | 179.99 | Desk with pull-out panel | Tables & desks | 0 | 0 | View Edit Delete |
| MALM | 199.99 | Desk | Tables & desks | 0 | 0 | View Edit Delete |

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

Showing filter that doesn't have any records (with an appropriate message)

Checklist Items (1)

#4 Demonstrate a filter that doesn't have any records (should show an appropriate message)

| Name | Price | Type Name | Category Path | Stock | is_api | Actions |
|------------|---------|------------------------------------|------------------|-------|--------|---|
| MATCH-SPEL | 200.99 | Gaming chair, Bernstad black | Chairs | 49 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LAGAN | 299.00 | Fridge with freezer compartment | Kitchen systems | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| STÄRMSÄTTA | 2699.00 | French door refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FÄRSKHEIT | 1199.00 | Bottom-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SUPERKALL | 1399.00 | Built-in, panel-ready refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| VÄLGRUNDAD | 1299.00 | Bottom-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| ÖVERSKÄDDA | 2999.00 | French door refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 195.00 | Top cab f fridge/freezer w push-up | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 133.00 | Top cab f fridge/freezer w 2 doors | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| PRYSBAR | 829.00 | Top-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |

Showing each list item with a link of single view, edit, and delete

Checklist Items (1)

#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)

Task #2 - Points: 1

Text: Screenshots of the list page code

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Show the filter/sort form generation |
| #2 | 1 | Show the DB query and how the filter/sort is handled (including the restriction on the limit field) |
| #3 | 1 | Show how the output is generated and displayed |
| #4 | 1 | Show any restrictions like role guard or login checks |
| #5 | 1 | Include ucid/date comments for each code screenshot |
| #6 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```
sanitizers.php 006_create_treble_products.sql Products list_products.php M db_helpers.php edit_product.php create_product.php M 007_alter_table_products_js_api.sql  
public_html > Project > admin > list_products.php > ..  
10 // +yc73 4/12/23 4/1 second ago + uncommitted changes  
11 //build search form  
12 $form = [  
13     ["type" => "text", "name" => "name", "placeholder" => "Name", "label" => "Product Name", "include_margin" => false],  
14     ["type" => "number", "name" => "price_min", "placeholder" => "Min Price", "label" => "Min Price", "step" => "0.01", "include_margin" => false],  
15     ["type" => "number", "name" => "price_max", "placeholder" => "Max Price", "label" => "Max Price", "step" => "0.01", "include_margin" => false],  
16     ["type" => "text", "name" => "typeName", "placeholder" => "Type", "label" => "Type", "include_margin" => false],  
17     ["type" => "text", "name" => "categoryPath", "placeholder" => "Category", "label" => "Category", "include_margin" => false],  
18 ]
```

Showing the filter/sort form generation

Checklist Items (1)

#1 Show the filter/sort form generation

Showing the DB query and how the filter/sort is handled (1)

Checklist Items (1)

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

```

17 //empty spruce_max & spruce_min =>0
18 spruce_min = 0;
19 spruce_max = 0;
20 spruce_min = min(spruce_min, spruce_max);
21 spruce_max = max(spruce_min, spruce_max);
22 spruce_min = spruce_min - 1;
23 spruce_max = spruce_max + 1;
24 spruce_min = max(spruce_min, 0);
25 spruce_max = min(spruce_max, 100);
26 spruce_min = max(spruce_min, 0);
27 spruce_max = min(spruce_max, 100);
28 spruce_min = max(spruce_min, 0);
29 spruce_max = min(spruce_max, 100);
30 spruce_min = max(spruce_min, 0);
31 spruce_max = min(spruce_max, 100);
32 spruce_min = max(spruce_min, 0);
33 spruce_max = min(spruce_max, 100);
34
35 //product type
36 $type_name = $_GET['type'];
37 if ($type_name) {
38     $query .= " AND type_name = '$type_name'";
39 }
40
41 //category
42 $category_name = $_GET['category'];
43 if ($category_name) {
44     $query .= " AND category_name = '$category_name'";
45 }
46
47 //sort
48 $sort_by = $_GET['sort'];
49 if ($sort_by) {
50     $query .= " ORDER BY $sort_by $sort_order";
51 }
52
53 //filter
54 $filter = $_GET['filter'];
55 if ($filter) {
56     $query .= " WHERE $filter";
57 }
58
59 //limit
60 $limit = $_GET['limit'];
61 if ($limit) {
62     $query .= " LIMIT $limit";
63 }
64
65 //execute query
66 $result = $db->query($query);
67
68 //check if result
69 if ($result) {
70     $products = $result->fetch_all('assoc');
71 }
72
73 //return products
74 return $products;
75 }
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173

```

Showing the DB query and how the filter/sort is handled (2) (and the limit field)

Checklist Items (0)

```

public_html > Project > admin > list_products.php > ...
126
127 //+ yC73 +/
128 /* 4/12/23 */
129 $db = getDB();
130 $stmt = $db->prepare($query);
131 $results = [];
132 try {
133     $stmt->execute($params);
134     $r = $stmt->fetchAll();
135     if ($r) {
136         $results = $r;
137     }
138 } catch (PDOException $e) {
139     error_log("Error fetching stocks ". var_export($e, true));
140     flash("An unhandled error occurred", "danger");
141 }
142
143
144 $stable = [
145     "data" => $results,
146     "/>title" => "Latest Stocks",
147     "/>ignored_columns" => ["id", "api_id", "measurement", "image", "contextualImageUrl", "imageAlt", "url"],
148     "/>view_url" => get_url("admin/view_product.php"),
149     "/>edit_url" => get_url("admin/edit_product.php"),
150     "/>delete_url" => get_url("admin/delete_product.php")
151 ];
152
153 <div class="container-fluid">
154     <div class="list-products-title">
155         <h3>List Products</h3>
156     </div>
157     <div class="list-products-container">
158         <form method="GET">
159             <div class="row mb-3" style="align-items: flex-end;">
160                 <?php foreach ($form as $k => $v) : ?>
161                     <div class="col">
162                         <?php render_input($v); ?>
163                     </div>
164                 <?php endforeach; ?>
165             </div>
166             <?php render_button("text" => "Search", "type" => "submit", "text" => "Filter"); ?>
167             <a href="?clear" class="btn btn-secondary">Clear</a>
168         </form>
169
170         <div class="container-fluid list-products">
171             <?php render_table($stable); ?>
172         </div>
173     </div>
174 </div>

```

Showing how the output is generated and displayed

Checklist Items (1)

#3 Show how the output is generated and displayed

```
public_html > Project > admin > 📁 list_products.php > ...
    You, 39 seconds ago | 1 author (You)
1  <?php
2  //note we need to go up 1 more directory
3  require(__DIR__ . "/../../../../partials/nav.php");
4
5  if (!has_role("Admin")) {
6      flash("You don't have permission to view this page", "warning");
7      die(header("Location: $BASE_PATH" . "/home.php"));
8  }
9
10 /* yc73 4/12/23 */
11 //build search form
12 $form = [
13
14     ["type" => "text", "name" => "name", "placeholder" => "Name", "label" => "Product Name"]
15
16     ["type" => "number", "name" => "price_min", "placeholder" => "Min Price", "label" => "M
```

Showing a role guard

Checklist Items (1)

#4 Show any restrictions like role guard or login checks

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Provide the high-level steps of how the filter/sorting works and how data is displayed |
| #2 | 1 | Summarize what you're showing on the screen |
| #3 | 1 | Mention your design/style choice |
| #4 | 1 | Mention which users can interact with the view, edit, and delete links |

Response:

The page starts by checking the user's role to see if they are an admin. If they are not an admin they are redirected to the home page.

The filter/sorting is made and displayed as a form at the top of the page. It has input fields (to search) for product name, product range, type, category, to sort (based on date created or modified, name, or price), and to limit (the number of results). There is also a "filter" button so users can confirm their choices and a "clear" button to remove any applied filters/sorting.

Upon loading the page or whenever the user submits the form by clicking "Filter", with a get request, it begins to

construct a query based on filter/sort criteria.

The "query" starts with a SELECT statement - to select the columns id, api_id, name, price, measurement, typeName, image, contextualImageURL, imageAlt, url, categoryPath, stock, and is_api - from the table Products and a WHERE 1=1 (to append AND).

A session key is also made and set to the current script "\$_SERVER["SCRIPT_NAME"]".

- If the clear button gets pressed, it sends "clear" with the get request. Then, the session data is deleted and the "clear" parameter is removed from GET.
- But, if there's no clear it proceeds to attempt to load the session and put it into "session_data". Next, if there are no elements in the GET array and the "session_data" is set, it puts that data into the GET array.
- Then, GET is checked again to see if the session data was successfully put into it. If there are elements in the GET array, it saves them in a session and grabs the keys from the GET array. A foreach loop iterates over the form and checks if each input field "name" (from the form) exists in "keys" (from the GET array). If it does, then the value for that field is updated with the value from the GET.

Continuing with the building of the query:

- For each filter/sort form field, it retrieves the data inputted by the user (for name, price (min and max), type, category, sort, and order) from the GET method. Then, it sets it to corresponding variables.
- If the variable is not empty, it proceeds to append the condition to the query (with an AND operator to filter the records based on that condition). A LIKE is used for product name, type, and category to specify a pattern to search for.
- For sort, the value for "sort" determines what column the results will be sorted by. But, if the value is invalid or not specified it will default to the "created" date.
- Similarly for order, the value for "order" determines in what order the results are going to be sorted (ascending or descending). If invalid or not specified, it defaults to descending.
- Then, the values for sort and order are appended to the query at the end, after the filter value(s) with ORDER BY. For limit, the value for "limit" determines how many results will appear on the page, if it's not specified, under 1, or over 100, then it defaults to 10 results. Then, the limit is appended to the end of the query after sort/order with LIMIT.
- Now that the query (for the entered filter/sort) has been put together it gets prepared and executed. The products (results) are rendered in a table only showing certain columns (name, price, type, category, stock, and is_api) because I wanted it to be more of a quick overview with only the important information regarding the store items. At the end of each product is also a view, edit, and delete button that can only be seen and used by admins because I don't want customers changing information on products.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Include the heroku prod link for this page |
| #2 | 1 | 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 |

URL #1

<https://github.com/yaneliii/vc73-it202-008/pull/53>

<https://github.com/yaneliii/yc73-it202-008/pull/53>

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/54>

URL #3

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/list_products.php

View Details Page (single entity) (1 pt.)

[^ COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the details page

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Entity should be fetch by id (via the url) |
| #2 | 1 | A missing id should redirect back to the list page with an applicable message |
| #3 | 1 | Design/Style should be considered (i.e., bootstrap, custom css, etc) |
| #4 | 1 | Data shown should be more detailed/inclusive than the summary view |
| #5 | 1 | 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) |
| #6 | 1 | 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) |
| #7 | 1 | Make sure the heroku dev url is visible in the address bar |
| #8 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a web browser window with multiple tabs open. The active tab displays the product details for a 'MATCHSPEL' gaming chair. The URL in the address bar is https://yc73-it202-008-dev-2cc66975f868.herokuapp.com/Project/admin/view_product.php?id=393. The page header includes links for 'Admin', 'Product Management', and 'Logout'. Below the header, there's a 'Back' button and the text 'Product: MATCHSPEL'. A large image of the chair is centered, followed by its name 'MATCHSPEL' and a table of specifications: Name: MATCHSPEL, Price: 289.99, Measurement: 26 x 52, Stock: 49.



Type: Gaming chair, Bornstad black
Category: Chairs
URL: https://www.ikea.com/us/en/p/matchspel-gaming-chair-bornstad-black-40507805/

Edit

Delete

Showing single entity view page: entity id in the url, data is more detailed, link to edit, link to delete

Checklist Items (4)

#1 Entity should be fetch by id (via the url)

#4 Data shown should be more detailed/inclusive than the summary view

#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)

#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)

The screenshot shows a web browser window with the following details:

- Tab Bar:** Calendar, yc73-it202-008-dev · GitHub, yc73-it202-008-dev-2cca · GitHub, MATCHSPEL gaming chair.
- Address Bar:** https://yc73-it202-008-dev-2cca6975f868.herokuapp.com/Project/admin/view_product.php?id=393
- Header:** Profile, Admin, Product Management, Logout.
- Main Content:** A large image of a black gaming chair with a mesh backrest and red accents on the base and headrest. To the right, there is a sidebar with the title "MATCHSPEL" and a text input field containing "Name: MATCHSPEL".

Showing entity being fetched by id (via the url)

Checklist Items (1)

#1 Entity should be fetch by id (via the url)

The screenshot shows a web application interface for managing products. At the top, there's a navigation bar with links like 'Add', 'Home', 'Profile', 'Admin', 'Product Management', and 'Logout'. Below the navigation, a pink error message box displays the text 'Invalid id passed'. The main content area is titled 'List Products'. It features a table with the following columns: Product Name, Min Price, Max Price, Type, Category, Sort, Order, and Limit. The 'Actions' column includes buttons for View, Edit, and Delete. The table lists various products such as MATCHSPEL, SEKTION, TILLREDA, LAGAN, FRYSBAR, and others, along with their respective details like price, category, and stock levels.

| name | price | typeName | categoryPath | stock | is_api | Actions |
|--------------|---------|------------------------------------|------------------|-------|--------|---|
| MATCHSPEL | 289.99 | Gaming chair, Bomstad black | Chairs | 49 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 185.00 | Top cabinet for fridge/freezer | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 185.00 | Top cab f fridge/freezer w push-op | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 133.00 | Top cab f fridge/freezer w 2 doors | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| TILLREDA | 119.00 | Refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LAGAN | 749.00 | Top-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FRYSBAR | 829.00 | Top-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LAGAN | 299.00 | Fridge with freezer compartment | Kitchen systems | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| ÖVERSKÄDLIG | 2999.00 | French door refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| VÄLGRUNDAD | 1299.00 | Bottom-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SUPERKALL | 1399.00 | Built-in, panel-ready refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FÄRSKHET | 1199.00 | Bottom-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| STJÄRNSTATUS | 2699.00 | French door refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| Tablee | 130.00 | Indoor table | Side & lamp | 25 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |

Showing how a missing id redirects user back to the list page with a message

Checklist Items (1)

#2 A missing id should redirect back to the list page with an applicable message

Task #2 - Points: 1

Text: Screenshots of the details page code

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Show how id is fetched |
| #2 | 1 | Show the DB query to get the record |
| #3 | 1 | Show the code related to presenting the data and showing the links |
| #4 | 1 | Include uid/date comments for each code screenshot |
| #5 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large

A screenshot of a code editor showing a PHP file named `view_product.php`. The code is used to fetch a product record by its ID from a database table named `Products`. It uses prepared statements and handles exceptions. The code is annotated with comments explaining the logic.

```
public_html > Project > admin > view_product.php > view_product.php M
  ↵ 10
  11  <?php
  12  /* yc73 4/12/23 */
  13  /* fetching id */
  14  $id = $_GET['id', -1, false];
  15
  16
  17  $product = [];
  18  if ($id > -1) {
  19      /* yc73 4/12/23 */
  20      //fetch
  21      $db = getDB();
  22      $query = "SELECT id, api_id, name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock, is_api, created, modified FROM 'Products' WHERE id = :id";
  23      try {
  24          $stmt = $db->prepare($query);
  25          $stmt->execute([':id' => $id]);
  26          $r = $stmt->fetch();
  27          if ($r) {
  28              $product = $r;
  29          }
  30      } catch (PDOException $e) {
  31          error_log("Error fetching record: " . var_export($e, true));
  32          flash("Error fetching record", "danger");
  33      }
  34  } else {
  35      flash("Invalid id passed", "danger");
  36      die(header("Location: " . get_url('admin/list_products.php')));
  37  }
  38  foreach ($product as $key => $value) {
  39      if (is_null($value)) {
  40          $product[$key] = 'N/A';
  41      }
  42  } <- #3B-42 foreach ($product as $key => $value)
```

Showing how id is fetched and the DB query to get the record

Checklist Items (2)

#1 Show how id is fetched

#2 Show the DB query to get the record

A screenshot of a web browser displaying a product gallery view. The page shows a large image at the top, followed by a carousel of smaller images. Below the images is a table with columns for Name, Price, Measurement, and Stock. The HTML code for this section is visible in the browser's developer tools.

```
public_html > Project > admin > view_product.php > ...
  ↵ 10
  11  <div class="container-fluid viewProdWhole">
  12  <div class="row align-items-center justify-content-between">
  13  <div href="#">php echo get_url('admin/list_products.php'); </div> <div>View Prod Back <a href="#">Back</a>
  14
  15  <div class="container-fluid viewProdContent">
  16  <div class="viewProd-Header">Products  <a href="https://yc73.com/trytrials/icon/original/988/620/986/Screenshot_2019-06-05_at_1-26-32_0h.jpg" -->
  17  <div class="card ex-autor" style="width: 60rem;" -->
  18
  19  <!-- yc73 4-05-23 -->
  20  <div class="row align-items-0">
  21  <div class="col-0">
  22      <div id="carousel" class="gallery-carousel pt-3 bg-light w-100 mx-auto">
  23          <div class="container">
  24              <div data-bs-interval="false" data-bs-ride="false" data-bs-pause="Never">
  25                  <div class="carousel-indicators">
  26                      <button type="button" data-bs-target="#CarouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
  27                      <button type="button" data-bs-target="#CarouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>
  28                  </div>
  29                  <div class="carousel-inner">
  30                      <div class="carousel-item active">
  31                           <div>w-100 alt="..."</div>
  32                      <div class="carousel-item">
  33                           <div>w-100 alt="..."</div>
  34                      </div>
  35                  </div>
  36                  <button class="carousel-control-prev" type="button" data-bs-target="#CarouselExampleIndicators" data-bs-slide="prev">
  37                      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  38                      <span class="visually-hidden">Previous</span>
  39                  </button>
  40                  <button class="carousel-control-next" type="button" data-bs-target="#CarouselExampleIndicators" data-bs-slide="next">
  41                      <span class="carousel-control-next-icon" aria-hidden="true"></span>
  42                      <span class="visually-hidden">Next</span>
  43                  </button>
  44          </div>
  45      </div>
  46      <div class="col-12 align-0" style="margin-left: -60px;">
  47          <table border="1" style="width: 100%; border-collapse: collapse; font-size: small; text-align: center; border: none; border-collapse: collapse; margin-bottom: 10px;">
  48              <tr>
  49                  <th>Name</th>
  50                  <th>Price</th>
  51                  <th>Measurement</th>
  52                  <th>Stock</th>
  53              </tr>
  54              <tr>
  55                  <td>Name</td>
  56                  <td>Price</td>
  57                  <td>Measurement</td>
  58                  <td>Stock</td>
  59              </tr>
  60          </table>
  61      </div>
  62  </div>
  63  </div>
```

Showing code related to presenting the data and showing the links

Checklist Items (1)

#3 Show the code related to presenting the data and showing the links

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Provide the high-level steps for handling the DB lookup and presenting the data |

Response:

On the list page, each product row has a "view" button associated with each product. When a user clicks that button it gets the (view_product.php) url along with the product "id" - (included at the end of the url) to get that specific page.

The view page retrieves the product id from the GET method and stores it in a variable "id". If the value of "id" was LESS than -1 then it redirects the user back to the list page and flashes a message saying it was invalid. If the "id" value was GREATER than -1, it can proceed to fetch the related product data from the db table.

- It begins making the query by starting it with the SELECT statement to select the columns (id, api_id, name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock, is_api, created, modified) from the Products table with a WHERE the "id" matches the "id" we grabbed earlier.
 - This ensures the data grabbed belongs to that specific product since every id is unique.
 - After preparing the query it can execute it and fetch the product.

- Once it has the product details from the db, it iterates through each key and checks if the value is null. If it is null, it assigns it with the string "N/A" so data doesn't appear as blank.

Then it uses html to render the product photos and details as a list, with the name of the column and its corresponding value. For example, Name: MALM, Price: \$165.99, Measurement: 80 x 60, Stock: 10, Type: Indoor desk brown, Category: Desks, Url: <https://www.ikea.com/.../>. There are also edit and delete buttons at the bottom so admins to perform other actions on that product.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking.

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Include the heroku prod link for this page |
| #2 | 1 | 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 |

URL #1

<https://github.com/yanelili/yc73-it202-008/pull/54>

URL #2

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/view_product.php?id=1

Edit Data Page (2 pts.)

^ COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots of the edit page

Checklist

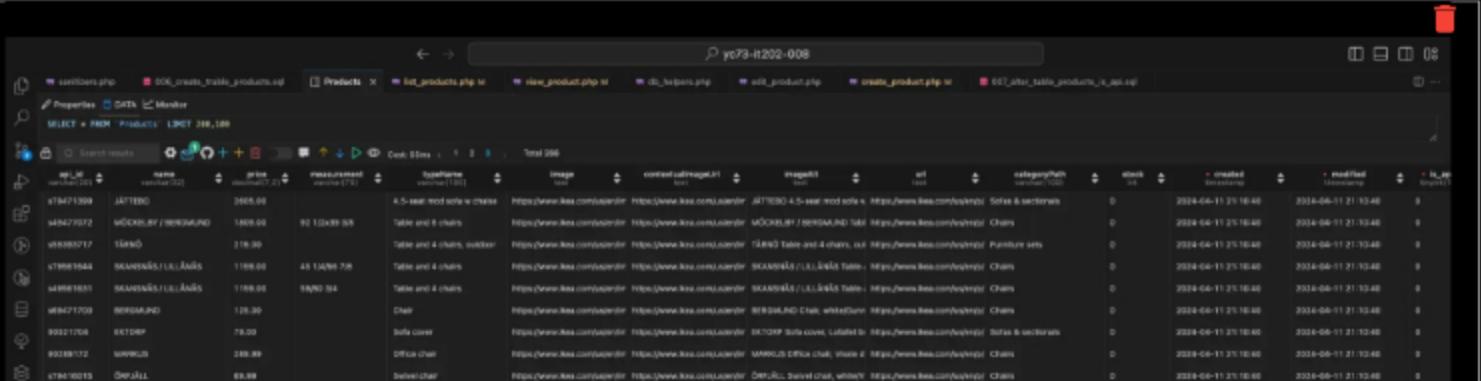
*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Show before and after screenshots of data you'll edit |
| #2 | 1 | Show examples of validation messages |
| #3 | 1 | Show an example of successful edit messages |
| #4 | 1 | Design/Style should be considered (i.e., bootstrap, custom css, etc) |
| #5 | 1 | Make sure the heroku dev url is visible in the address bar |
| #6 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Showing product data in table BEFORE edit

Checklist Items (1)

#1 Show before and after screenshots of data you'll edit

Calendar | yc73-it202-008-dev-2cc6975f668.herokuapp.com | yc73-4202-008-dev-2cc6975f668 | MATCHSPEL_gaming_chair | matchs spel-gaming-chair-1 | matchs spel-gaming-chair-1 | +

https://yc73-it202-008-dev-2cc6975f668.herokuapp.com/Project/admin/edit_product.php?id=314

Logout

Product Management

Back

Edit Store Item

Product Name:

Product Price:

Product Measurement:

Product Type (name):

Product Image:

Product Continue Image:

Product Image Alt:

Product URL:

Product Category:

Product Stock (number):

Showing data (on edit page) BEFORE edit

Checklist Items (1)

#1 Show before and after screenshots of data you'll edit

Showing form with wrong data (to get error messages)

Checklist Items (1)

#2 Show examples of validation messages

Showing examples of validation messages for each field

Checklist Items (1)

#2 Show examples of validation messages

A screenshot of a web browser window displaying a product edit form. The URL is https://yc73-it202-008-dev-2cc6975f868.herokuapp.com/Project/admin/edit_product.php?id=314. The form contains the following fields:

- Product Name: REVOLVERRRRRR
- Product Price: 799.00
- Product Measurement: 91.18 x 25.58
- Product Type (name): 3-seat conversation set, outdoor armchairs EDIT
- Product Image: https://www.reis.com/cdn/images/products/reviews/3-seat-conversation-set-outdoor-armchairs-furniture-dark-gray_1140171_pcl1021_stg.jpg
- Product Contextual Image: https://www.reis.com/cdn/images/products/reviews/3-seat-conversation-set-outdoor-armchairs-furniture-dark-gray_1140171_pcl1021_stg.jpg
- Product Image Alt: REVOLVER 3-seat conversation set, outdoor armchairs, dark gray EDIT
- Product Uf: https://www.reis.com/cdn/images/products/reviews/3-seat-conversation-set-outdoor-armchairs-furniture-dark-gray_1140171_pcl1021_stg.jpg
- Product Category: Outdoors EDIT
- Product Stock (number): 10

The "Update" button is at the bottom of the form.

Showing form with correct data

Checklist Items (1)

#3 Show an example of successful edit messages

A screenshot of a web browser window displaying a product edit form. The URL is https://yc73-it202-008-dev-2cc6975f868.herokuapp.com/Project/admin/edit_product.php?id=314. The form is identical to the one above, but it includes a green header bar with the text "Updated record". The rest of the form and its fields are the same as in the previous screenshot.

Showing successful edit message

Checklist Items (1)

#3 Show an example of successful edit messages

| Screenshot of a successful edit message in a database table. | | | | | | | | | | | | | | | | | |
|--|---------------------|---------|-------------|--------------------------------------|---|---|-----------------------|-----|---------------------|---------------------|---------|----------|---------|-----------|------------|--|--|
| id | name | price | measurement | type | image | content | ImageAlt | url | categoryPath | stock | created | modified | version | isDeleted | isArchived | | |
| 479471093 | JETT93 | 2000.00 | cm²/m² | 4.5-seat mod. sofa in chaise | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../JETT93-4.5-seat-mod-sofa-in-chaise.html | Stools & seatcushions | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 479471072 | MÖÖKELBY / BERGMUND | 1000.00 | 92x70x90 cm | Table and 4 chairs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../MÖÖKELBY / BERGMUND-table-and-4-chairs.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 509993717 | TÄRNÖ | 210.00 | | Table and 4 chairs, outdoor | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../TÄRNÖ-Table-and-4-chairs-outdoor.html | Furniture sets | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 579591644 | SKÅNHÅS / ULJÄRÅS | 1150.00 | 40x50x70 cm | Table and 4 chairs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../SKÅNHÅS / ULJÄRÅS-Table-and-4-chairs.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 549591631 | SKÅNHÅS / ULJÄRÅS | 1150.00 | 59x80 cm | Table and 4 chairs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../SKÅNHÅS / ULJÄRÅS-Table-and-4-chairs.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 59471106 | BERÖMND | 125.00 | | Chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../BERÖMND-Chair-white.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 50321794 | EKTORP | 75.00 | | Sofa cover | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../EKTORP-Sofa-cover-Luftslott-le.html | Stools & seatcushions | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 50288171 | MÄRKLIG | 288.00 | | Office chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../MÄRKLIG-Office-chair-Växle-le.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 579471010 | ÖRTJÄLL | 88.00 | | Swivel chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../ÖRTJÄLL-Swivel-chair-white.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 50465787 | LÖDULLEN | 133.00 | | Skidswivel support | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../LÖDULLEN-Skidswivel-support.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 50423336 | FLINTAN | 75.00 | | Office chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../FLINTAN-Office-chair-Ljunge.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 503417469 | POÄNG | 44.00 | | Kid's armchair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../POÄNG-Kid-s-armchair-white.html | Chairs | 0 | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | | | | | |
| 547717945 | ELRKJERBERG | 109.00 | 20x30x70 cm | Armchair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../ELRKJERBERG-Armchair-white.html | Outdoor armchairs | 46 | 2024-04-10 08:02:11 | 2024-04-10 08:03:00 | 0 | | | | | | |
| 485773883 | UTSAED | 68.00 | 14x18x30 cm | Car seat | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../UTSAED-Car-seat-on-legs-white.html | Pet accessories | 14 | 2024-04-17 12:39:43 | 2024-04-17 12:40:43 | 0 | | | | | | |
| 66646831 | REVKÄRÄNNÄRÖ | 750.00 | 83x18x25 cm | 3-seat conversation set, outdoor | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../REVKÄRÄNNÄRÖ-3-seat-conversation-set-outdoor.html | Editt | 16 | 2024-04-18 21:19:20 | 2024-04-18 21:47:58 | 1 | | | | | | |
| 214934833 | TESSAMÄRAN | 16.00 | 14x18x20 cm | Side table, radioboard unit | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../TESSAMÄRAN-Side-table.html | Tables | 26 | 2024-04-16 21:27:27 | 2024-04-16 21:27:27 | 1 | | | | | | |
| 50523239 | PERUJUHAN | 40.00 | 20x30x17 cm | Bedbox with storage, white | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../PERUJUHAN-Bedbox-with-storage.html | Chairs | 24 | 2024-04-16 09:02:23 | 2024-04-16 09:02:23 | 1 | | | | | | |
| 5238367 | EVIK | 109.00 | 50x60 cm | Log table | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../EVIK-log-table.html | Tables | 16 | 2024-04-16 09:23:35 | 2024-04-16 09:23:35 | 1 | | | | | | |
| 201654332 | KVÄLLA | 173.00 | 27x10x70 cm | High cabinet with doors, white | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../KVÄLLA-high-cabinet-with-2-doors.html | Cabinets & shelves | 12 | 2024-04-16 14:05:12 | 2024-04-16 14:05:12 | 1 | | | | | | |
| 667644332 | TÄVELS | 130.00 | 54x18x26 cm | Indoor table | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../TÄVELS-indoor-table.html | Side & lamp | 26 | 2024-04-16 10:33:02 | 2024-04-16 10:33:02 | 1 | | | | | | |
| 20522287 | STJÄRNGÅTTUS | 2090.00 | 22x4 m | French door refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../STJÄRNGÅTTUS-french-door-refrigerator.html | Appliances | 9 | 2024-04-16 16:05:00 | 2024-04-16 16:05:00 | 1 | | | | | | |
| 50462155 | FÄRGKÄTT | 1130.00 | 90x4 m | Glass-front refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../FÄRGKÄTT-glass-front-refrigerator.html | Appliances | 9 | 2024-04-16 16:05:00 | 2024-04-16 16:05:00 | 1 | | | | | | |
| 50462156 | SURFRÄKK | 1390.00 | 6x6 m | Glass-in, panel-ready refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../SURFRÄKK-glass-in-panel-ready-refrigerator.html | Appliances | 9 | 2024-04-16 16:05:00 | 2024-04-16 16:05:00 | 1 | | | | | | |
| 50462158 | VÄLGRÄKKAD | 1190.00 | 59x60 cm | Glass-front refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../VÄLGRÄKKAD-glass-front-refrigerator.html | Appliances | 9 | 2024-04-16 16:05:00 | 2024-04-16 16:05:00 | 1 | | | | | | |
| 50462154 | ÖNDRÄKKAD | 2090.00 | 32x60 cm | French door refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../ÖNDRÄKKAD-french-door-refrigerator.html | Appliances | 9 | 2024-04-16 16:05:00 | 2024-04-16 16:05:00 | 1 | | | | | | |
| 504601110 | LAGAN | 209.00 | 4.0 m | Fridge with freezer compartments | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../LAGAN-fridge-with-freezer-compartment.html | Kitchen systems | 5 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 50462168 | FEJKA | 839.00 | 13.0 m | Top-freezer refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../FEJKA-top-freezer-refrigerator.html | Appliances | 9 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 50521503 | LAGAN | 743.00 | 13.0 m | Top-freezer refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../LAGAN-top-freezer-refrigerator.html | Appliances | 9 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 504627053 | TILLBEDA | 510.00 | 2 m | Refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../TILLBEDA-refrigerator.html | Appliances | 9 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 50543874 | EDITION | 133.00 | 30x30x20 | Top rack / bridge-freezer w/ 2 doors | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../EDITION-top-rack-bridge-freezer-w-2-doors.html | Kitchen cabinets | 0 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 5053948205 | SEKTION | 185.00 | 36x30x10 | Top rack / bridge-freezer w/ 2 doors | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../SEKTION-top-rack-bridge-freezer-w-2-doors.html | Kitchen cabinets | 0 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 5053948106 | SEKTION | 185.00 | 36x30x10 | Top cabinet for bridge-freezer | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../SEKTION-top-cabinet-for-bridge-freezer.html | Kitchen cabinets | 0 | 2024-04-16 10:35:00 | 2024-04-16 10:35:00 | 1 | | | | | | |
| 40507636 | MATCHSPEL | 299.99 | 20 x 12 | Gaming chair, Romeo black | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/vir.../MATCHSPEL-gaming-chair-Romeo-black.html | Chairs | 49 | 2024-04-16 10:35:17 | 2024-04-16 10:35:17 | 1 | | | | | | |

Showing product data in table AFTER edit

Checklist Items (1)

#3 Show an example of successful edit messages

Task #2 - Points: 1

Text: Screenshots of edit page code

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Form should have correct data types for each property being requested |
| #2 | 1 | Form should have correct validation for each field (HTML, JS, and PHP) |
| #3 | 1 | Successful edit should have a user-friendly message |

- #3 1 Any errors should have user-friendly messages
- #4 1 Include any other rules like role guards and login checks
- #5 1 Include ucid/date comments for each code screenshot
- #6 1 Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
public_html>Project>admin>edit_product.php>script>validate
  ...
  } <- #62-79 if ($id > -1)
else {
    flash("Invalid id passed", "danger");
    die(header("Location: " . get_url("admin/list_products.php")));
}

/* jc23 4/14/23 */
/* edit product form */
if($product) {
    $form = [
        // ...
        ...
    ];
    ...
}

$keys = array_keys($product);
error_log("keys " . var_export($keys, true));
foreach ($form as $k => $v) {
    if (in_array($v["name"], $keys)) {
        error_log("IN ARRAY");
        $form[$k]["value"] = $product[$v["name"]];
        error_log("Value: " . var_export($v, true));
    }
} <- #105-109 if (in_array($v["name"], $keys))
} <- #110-110 foreach ($form as $k => $v)
/*error_log("Form full data " . var_export($form, true));*/
} <- #112 if($product)

// 1000 handle manual create product
?>

<div class="container editProd-form shadow"> <!-- card d-flex justify-content-center -->
    <div>
        <a href="php echo get_url("admin/list_products.php"); ?" class="editProd-back btn btn-secondary">Back</a>
    </div>
    <h3 class="editProd_title">Edit Store Item</h3>
    <form onsubmit="return validate(this)" method="POST">
        <input type="text" name="name" value="Product API ID" placeholder="Product API ID" required>
        <input type="text" name="name" value="Product Name" placeholder="Product Name" required>
        <input type="number" name="price" value="100" placeholder="Product Price" required>
        <input type="text" name="measurement" value="100" placeholder="Product Measurement" required>
        <input type="text" name="typeName" value="Electronics" placeholder="Product Type (name)" required>
        <input type="text" name="image" value="product1.jpg" placeholder="Product Image" required>
        <input type="text" name="contextualImage" value="productContextualImage" placeholder="Product Contextual Image" required>
        <input type="text" name="imageAlt" value="Product Image Alt" placeholder="Product Image Alt" required>
        <input type="text" name="url" value="http://example.com/product" placeholder="Product Url" required>
        <input type="text" name="categoryPath" value="Electronics/Computers" placeholder="Product Category" required>
        <input type="number" name="stock" value="100" placeholder="Product Stock (number)" required>
    </form>

```

Showing form with correct data types for each property being requested

Checklist Items (1)

#1 Form should have correct data types for each property being requested

Showing form with javascript validation (1)

Checklist Items (1)

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

Showing form with javascript validation (2)

Checklist Items (1)

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

Showing form with php validation (1)

Checklist Items (1)

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

Showing form with php validation (2)

Checklist Items (1)

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

A screenshot of a code editor showing a PHP file named `edit_product.php`. The code handles POST requests to update products in a database. It includes validation for the `name` field, which is checked against the `$_POST` array. The code uses `var_export` to log variable values and `error_log` to log messages. It also includes a `try` block for database operations.

```
<?php
// yc73 4/14/23 /*
$Id = isset($_GET['id'], -1, false);

// handle protect fetch
if (!isset($_POST['keyword'])) {
    $action = $_POST['action'];
}

if (isset($_POST['name'])) {
    foreach ($_POST as $k => $v) {
        //yc73 4/14/23 /*
        error_log("cleared up POST: " . var_export($_POST, true));
        if (!in_array($k, ['id', 'name', 'price', 'measurement', 'typelabel', 'image', 'contextualImageTitle', 'imageAlt', 'url', 'categoryPath', 'stock'])) {
            unset($_POST[$k]);
        }
    }
    $quote = $_POST;
    error_log("cleared up POST: " . var_export($quote, true));
} // -- 423-16 foreach ($POST as $k => $v)
//insert data
$db = getDB();
$query = "UPDATE `Products` SET ";
$params = [];

//per record
foreach ($quote as $k => $v) {
    if ($params) {
        $query .= ", ";
    }
    //The value $v is trusted as this is a source of SQL injection
    $query .= "$k=$v";
    $params[] = $v;
}
// -- 437-44 foreach ($quote as $k => $v)

$query .= " WHERE id = ?"; //id
$params[0] = $Id;
error_log("Query: " . $query);
error_log("Params: " . var_export($params, true));

try {
    $stmt = $db->prepare($query);
    $stmt->execute($params);
    //yc73 4/14/23 /*
    // success message/
    flash("Updated record ", "success");
} catch (PDOException $e) {
    error_log("Something broke with the query" . var_export($e, true));
    flash("An error occurred", "danger");
}
} // -- 422-63 if (isset($_POST['name']))

$product = [];
if ($Id > -1) {
    //fetch
}
```

Showing successful message

Checklist Items (1)

#3 Successful edit should have a user-friendly message

A screenshot of a code editor showing a PHP file named `edit_product.php`. The code checks if the user has the 'Admin' role. If not, it displays a warning message and a header to redirect to the home page. The code uses `header` to set the location.

```
<?php
// note we need to go up 1 more directory
require(__DIR__ . "/../../../../partials/nav.php");

// yc73 4/14/23 /*
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: $BASE_PATH" . "/home.php"));
}

// -- 7>

<?php
```

Showing role guard

Checklist Items (1)

#5 Include any other rules like role guards and login checks

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|--|
| #1 | 1 | Show a before and after screenshot of the record |
| #2 | 1 | Note what differs |
| #3 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small Medium Large

| yc73-h202-008 | | | | | | | | | | | | | | | | | |
|---|------------------------|------------|-------------------------------|------------------------------------|---|---|--|---------------------------------|----------------------|-----------|-----|------|-------------------|------------|---------------------|---------------------|----------|
| Properties | | DATA | | Member | | | | | | | | | | | | | |
| SELECT * FROM `Products` LIMIT 300,1000 | | | | Products | | | | | | | | | | | | | |
| # | Name | CategoryID | Price | MeasurementUnit | MeasureUnitID | TypeName | Value1 | ImageURL | ContractSignatureURL | ImageEdit | Alt | Last | CategoryPath | CategoryID | Stock | Created | Modified |
| 1 | JÄTTERBO | 2405.00 | 4.5-kat. med säng och chaisse | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/JÄTTERBO/4.5-kat. med säng och chaisse | https://www.kea.com/kea/leider/JÄTTERBO/4.5-kat. med säng och chaisse | 4.5-kat. med säng och chaisse | | | | | | Sofas & sängar | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 2 | MÖCKELBY / HIRSHAMUND | 1600.00 | 92 120x89 88 | Tablet och 4 chairs | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/MÖCKELBY / HIRSHAMUND 88x | MÖCKELBY / HIRSHAMUND 88x | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 3 | TÅMÖ | 218.00 | Tablet och 4 chairs, matbord | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/TÅMÖ Tablet och 4 chairs, matbord | TÅMÖ Tablet och 4 chairs, matbord | Tablet och 4 chairs, matbord | https://www.kea.com/kea/leider/ | | | | | Puntbord matbord | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 4 | KÄRMEHÅL / LILLÄNNS | 1199.00 | 45 110x94 798 | Tablet och 4 chairs | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/KÄRMEHÅL / LILLÄNNS Tablet och 4 chairs | KÄRMEHÅL / LILLÄNNS Tablet och 4 chairs | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 5 | SKÄRMÅSILJU / LILLÄNNS | 1199.00 | 5980 384 | Tablet och 4 chairs | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/SKÄRMÅSILJU / LILLÄNNS Tablet och 4 chairs | SKÄRMÅSILJU / LILLÄNNS Tablet och 4 chairs | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 6 | BERGMUND | 119.00 | Chair | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/BERGMUND Chair, white/black | BERGMUND Chair, white/black | Chair | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 7 | EXT049 | 79.00 | Soft sofa cover | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/EXT049 Soft sofa cover, lotsöld si | EXT049 Soft sofa cover, lotsöld si | Soft sofa cover | https://www.kea.com/kea/leider/ | | | | | Sofas & värderörs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 8 | MÄRKLUS | 189.49 | Office chair | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/MÄRKLUS Office chair, white | MÄRKLUS Office chair, white | Office chair | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 9 | DRÄJUL | 89.49 | Swivel chair | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/DRÄJUL Swivel chair, white | DRÄJUL Swivel chair, white | Swivel chair | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 10 | LEOKULLEN | 109.49 | Skyddställ support | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/LEOKULLEN Skyddställ support | LEOKULLEN Skyddställ support | Skyddställ support | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 11 | FILMAN | 79.49 | Office chair | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/FILMAN Office chair, beige | FILMAN Office chair, beige | Office chair | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 12 | POÄNG | 44.99 | Kid's armchair | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/POÄNG Kid's armchair, birch v | POÄNG Kid's armchair, birch v | Kid's armchair | https://www.kea.com/kea/leider/ | | | | | Chairs | 0 | 2024-06-11 21:18:48 | 2024-06-11 21:18:48 | |
| 13 | ULÄRKBECKS | 119.00 | 28 38x29 798 | Armchair | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/ULÄRKBECKS Armchair, raffar | ULÄRKBECKS Armchair, raffar | https://www.kea.com/kea/leider/ | | | | | Outdoor armchairs | 49 | 2024-06-15 08:22:11 | 2024-06-15 14:06:29 | |
| 14 | UF940 | 99.00 | 14 18x20 798 | Garden | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/UF940 Garden | Garden | https://www.kea.com/kea/leider/ | | | | | Outdoors | 14 | 2024-06-17 13:55:43 | 2024-06-17 19:35:43 | |
| 15 | REVSÅB | 855.00 | 85 18x 51 1.8 x 29.50 | 3-seater conversation set, oak | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/REVSÅB 3-seater conversation set, oak | REVSÅB 3-seater conversation set, oak | https://www.kea.com/kea/leider/ | | | | | Outdoor sets | 210 | 2024-06-19 25:19:26 | 2024-06-19 21:13:38 | |
| 16 | TESSAMÄRIS | 19.00 | 14 18x20 798 | Side table, nordic oak | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/TESSAMÄRIS Side table | TESSAMÄRIS Side table | https://www.kea.com/kea/leider/ | | | | | Tables | 29 | 2024-06-19 25:27:27 | 2024-06-18 21:27:27 | |
| 17 | PERJUHAN | 49.99 | 39 38x 17 5/4 | Bench with storage, pine | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/PERJUHAN Bench with storage | PERJUHAN Bench with storage | https://www.kea.com/kea/leider/ | | | | | Chairs | 74 | 2024-06-19 05:02:23 | 2024-06-19 02:02:23 | |
| 18 | DÖR | 199.99 | 59 x 50 1/4 | Hipster | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/DÖR | DÖR | https://www.kea.com/kea/leider/ | | | | | Chairs | 19 | 2024-06-19 05:26:39 | 2024-06-19 05:17:17 | |
| 19 | HÄLSA | 179.00 | 27 112 x 74 5/8 | High cabinet with 2 doors, oak | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/HÄLSA High cabinet with 2 doors, oak | HÄLSA High cabinet with 2 doors, oak | https://www.kea.com/kea/leider/ | | | | | Cabinets & shelf | 12 | 2024-06-19 14:05:12 | 2024-06-19 14:05:12 | |
| 20 | TÄLLER | 180.00 | 14 18x 20 798 | Side table | www.kea.com/kea/leider/ | www.kea.com/kea/leider/täller | täller | www.kea.com/kea/leider/ | | | | | Tables | 29 | 2024-06-19 15:23:33 | 2024-06-19 15:23:33 | |
| 21 | STÄRMMÄSTUR | 2499.00 | 22.0 cu ft | French door refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/STÄRMMÄSTUR French door | STÄRMMÄSTUR French door | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 16:39:00 | 2024-06-19 16:39:00 | |
| 22 | HÄRSÄHET | 1199.00 | 13.4 cu ft | Bottom-freezer refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/HÄRSÄHET Bottom-freezer | HÄRSÄHET Bottom-freezer | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 16:38:00 | 2024-06-19 16:38:00 | |
| 23 | SUPERKULL | 1899.00 | 8.8 cu ft | Built-in, panel-ready cold special | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/SUPERKULL Built-in, panel-ready cold special | SUPERKULL Built-in, panel-ready cold special | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 16:38:00 | 2024-06-19 16:38:00 | |
| 24 | VÄLÖRÄKUDRO | 1299.00 | 19 cu ft | Bottom-freezer refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/VÄLÖRÄKUDRO Bottom-freezer | VÄLÖRÄKUDRO Bottom-freezer | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 16:38:00 | 2024-06-19 16:38:00 | |
| 25 | ÖVERKÄÄLIS | 2999.00 | 22 cu ft | French door refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/ÖVERKÄÄLIS French door | ÖVERKÄÄLIS French door | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 16:38:30 | 2024-06-19 16:38:30 | |
| 26 | PERJUHÄN | 299.00 | 4.2 cu ft | Fridge with freezer, composite | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/PERJUHÄN Fridge with freezer, composite | PERJUHÄN Fridge with freezer, composite | https://www.kea.com/kea/leider/ | | | | | Kitchen systems | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 27 | PERJUHÄN | 219.00 | 13.9 cu ft | Top-freezer refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/PERJUHÄN Top-freezer refrigerator | PERJUHÄN Top-freezer refrigerator | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 28 | PERJUHÄN | 149.00 | 13.9 cu ft | Top-freezer refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/PERJUHÄN Top-freezer refrigerator | PERJUHÄN Top-freezer refrigerator | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 29 | TILLERÖDA | 119.00 | 2 cu ft | Refrigerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/TILLERÖDA Refrigerator | TILLERÖDA Refrigerator | https://www.kea.com/kea/leider/ | | | | | Appliances | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 30 | SEXTION | 192.00 | 29x16x23 | Top-freezer refrigerator in 2.0 | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/SEXTION Top-freezer refrigerator in 2.0 | SEXTION Top-freezer refrigerator in 2.0 | https://www.kea.com/kea/leider/ | | | | | Kitchen cabinets | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 31 | SEXTION | 185.00 | 28x16x19 | Top-freezer refrigerator in 2.0 | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/SEXTION Top-freezer refrigerator in 2.0 | SEXTION Top-freezer refrigerator in 2.0 | https://www.kea.com/kea/leider/ | | | | | Kitchen cabinets | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 32 | SEXTION | 185.00 | 29x24x19 | Top-cabinet for fridgerator | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/SEXTION Top-cabinet for fridgerator | SEXTION Top-cabinet for fridgerator | https://www.kea.com/kea/leider/ | | | | | Kitchen cabinets | 0 | 2024-06-19 18:20:20 | 2024-06-19 18:20:20 | |
| 33 | MÄRTHESPEL | 266.39 | 38 x 52 | Swing chair, domed back | https://www.kea.com/kea/leider/ | https://www.kea.com/kea/leider/MÄRTHESPEL Swing chair, domed back | MÄRTHESPEL Swing chair, domed back | https://www.kea.com/kea/leider/ | | | | | Chairs | 49 | 2024-06-19 19:40:17 | 2024-06-19 19:40:17 | |

Showing product data in table BEFORE edit (original data as entered before any changes)

Checklist Items (1)

#1 Show a before and after screenshot of the record

| id | name | price | measurement | typeName | image | contextualImageId | imageAlt | url | categoryPath | stock | created | modified | lastEdit |
|-----------|----------------------|---------|-----------------|------------------------------------|---|---|--------------------|---|---------------------|---------------------|---------------------|----------|----------|
| 479871389 | JÄTTESO | 2899.00 | 4.8 cu ft | 4-seat sofa in chaise | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/JÄTTESO-4-seat-sofa-in-chaise--https://www.ikea.com/us/en/itm/ | Sofas & settees | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 469476702 | MÖCKELIG / BERGMUND | 1099.00 | 92.5x99.19 | Table and 6 chairs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/MÖCKELIG-BERGMUND-table--https://www.ikea.com/us/en/itm/ | Tables & chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 679999117 | TÅRNÖ | 219.00 | | Table and 4 chairs, outdoor | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/TÅRNÖ-Table-and-4-chairs-outdoor--https://www.ikea.com/us/en/itm/ | Furniture sets | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 479616444 | SKÄNNINGE / LILLÅNGS | 1199.00 | 40.5x60.78 | Table and 4 chairs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/SKÄNNINGE-LILLÅNGS-Table--https://www.ikea.com/us/en/itm/ | Tables & chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 479516101 | SKÄNNINGE / LILLÅNGS | 1199.00 | 5980.54 | Table and 4 chairs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/SKÄNNINGE-LILLÅNGS-Table--https://www.ikea.com/us/en/itm/ | Tables & chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 489471790 | BORGSUND | 139.00 | | Chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/BORGSUND-Chair-white/black--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 603817944 | EKTORP | 29.00 | | Glossy chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/EKTORP-Glossy-chair-white--https://www.ikea.com/us/en/itm/ | Chairs & settees | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 603881713 | MÅRKLÅS | 389.00 | | Office chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/MÅRKLÅS-Office-chair-Vinete--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 479416015 | ÖDEJÄLL | 69.00 | | Swivel chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/ÖDEJÄLL-Swivel-chair-white--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 504857877 | LÖKHJELLEN | 139.00 | | Griphead support | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/LÖKHJELLEN-Griphead-support--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 00402395 | FLYTKÄN | 79.00 | | Office chair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/FLYTKÄN-Office-chair-white--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 603817699 | POÄNG | 44.00 | | Kid's armchair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/POÄNG-Kid-s-armchair-white--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 2024-04-11 21:19:40 | 2024-04-11 21:10:40 | 0 | | |
| 20817046 | ULRIKSBERG | 109.00 | 26.5x39.78 | Armchair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/ULRIKSBERG-Armchair-rattan--https://www.ikea.com/us/en/itm/ | Outdoor armchairs | https://www.ikea.com/us/en/ | 46 | 2024-04-16 04:02:11 | 2024-04-16 14:09:39 | 0 | |
| 60873843 | UTSÅLD | 69.00 | 54.5x30.78 | Cat bed | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/UTSÅLD-Cat-house-on-legs--https://www.ikea.com/us/en/itm/ | Pet accessories | https://www.ikea.com/us/en/ | 14 | 2024-04-17 13:39:43 | 2024-04-17 13:39:43 | 0 | |
| 489464831 | TESSAMMANS | 799.00 | 83.1/8 x 26.5/8 | 3-seat conversation set, outdoor | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/TESSAMMANS-3-seat-conversation-set-outdoor--https://www.ikea.com/us/en/itm/ | Outdoor sofa EDIT | https://www.ikea.com/us/en/ | 16 | 2024-04-18 21:13:20 | 2024-04-18 21:17:58 | 1 | |
| 218948483 | TRÅSAMMANS | 58.00 | 14.5x10.78 | Side table, reddish brown/pink | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/TRÅSAMMANS-Side-table--https://www.ikea.com/us/en/itm/ | Tables & chairs | https://www.ikea.com/us/en/ | 26 | 2024-04-18 21:37:27 | 2024-04-18 21:37:37 | 1 | |
| 603881399 | PERJUHAN | 89.00 | 39.3/8 x 17.5/4 | Bench with storage, pine | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/PERJUHAN-Bench-with-storage--https://www.ikea.com/us/en/itm/ | Desks | https://www.ikea.com/us/en/ | 74 | 2024-04-18 09:02:23 | 2024-04-18 09:33:23 | 1 | |
| 1238467 | DESK | 199.00 | 80 x 43.14 | Big table | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/DESK-big-table--https://www.ikea.com/us/en/itm/ | Tables & chairs | https://www.ikea.com/us/en/ | 16 | 2024-04-18 09:33:38 | 2024-04-18 13:17:17 | 1 | |
| 20817048 | HÅLIGA | 179.00 | 27.1/2 x 78.3/8 | High cabinet with 2 doors, white | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/HÅLIGA-High-cabinet-with-2-doors--https://www.ikea.com/us/en/itm/ | Cabinet & shelf | https://www.ikea.com/us/en/ | 12 | 2024-04-18 14:06:12 | 2024-04-18 14:08:17 | 1 | |
| 603864322 | TÄBLA | 199.00 | 54.5x20.78 | Indoor table | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/TÄBLA-Indoor-table--https://www.ikea.com/us/en/itm/ | Side & temp. | https://www.ikea.com/us/en/ | 26 | 2024-04-18 19:33:03 | 2024-04-18 19:35:58 | 1 | |
| 20523287 | STÅRNSTATUS | 2999.00 | 23.6 cu ft | French door refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/STÅRNSTATUS-French-door-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 18:38:00 | 2024-04-18 18:39:51 | 1 | |
| 30802166 | FÄRSKAD | 1199.00 | 10.4 cu ft | Bottom-freezer refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/FÄRSKAD-Bottom-freezer-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:36:00 | 1 | |
| 103802146 | SUPERKALL | 1399.00 | 8.8 cu ft | Built-in, panel-ready refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/SUPERKALL-Built-in-panel-ready-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 70482158 | MÅLGRUNDAD | 1399.00 | 19 cu ft | Bottom-freezer refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/MÅLGRUNDAD-Bottom-freezer-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 603862154 | ÖVERSKÄLIGÅ | 2999.00 | 22 cu ft | French door refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/ÖVERSKÄLIGÅ-French-door-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 50481119 | LAGÅN | 299.00 | 4.0 cu ft | Fridge with freezer compartment | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/LAGÅN-Fridge-with-freezer-compartment--https://www.ikea.com/us/en/itm/ | Kitchen appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 60382148 | FRYSBAR | 899.00 | 13.9 cu ft | Top-freezer refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/FRYSBAR-Top-freezer-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 603881532 | LAGÅN | 799.00 | 13.9 cu ft | Top-freezer refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/LAGÅN-Top-freezer-refrigerator--https://www.ikea.com/us/en/itm/ | Appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 20847023 | TÅLLÅRA | 119.00 | 2 m² | Refrigerator | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/TÅLLÅRA-Refrigerator--https://www.ikea.com/us/en/itm/ | Kitchen appliances | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 603881374 | SEKTION | 139.00 | 30x34x30 | Top-cut 1-fridge/water w/ 2 drs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/SEKTION-Top-cut-1-fridgewater-w-2-drss--https://www.ikea.com/us/en/itm/ | Kitchen cabinets | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 459384405 | SEKTION | 185.00 | 36x34x10 | Top-cut 1-fridge/water w/ 2 drs | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/SEKTION-Top-cut-1-fridgewater-w-2-drss--https://www.ikea.com/us/en/itm/ | Kitchen cabinets | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 603881816 | SEKTION | 185.00 | 36x34x10 | Top-cutters for 1-fridge/water | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/SEKTION-Top-cutters-for-1-fridgewater--https://www.ikea.com/us/en/itm/ | Kitchen cabinets | https://www.ikea.com/us/en/ | 8 | 2024-04-18 19:35:00 | 2024-04-18 19:35:00 | 1 | |
| 40507488 | MATCHSPIL | 249.00 | 26 x 62 | Dining chair, black/marble black | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/MATCHSPIL-Dining-chair-black-marble-black--https://www.ikea.com/us/en/itm/ | Chairs | https://www.ikea.com/us/en/ | 48 | 2024-04-18 18:42:17 | 2024-04-18 18:42:17 | 1 | |

| Showing product data in table AFTER edit (1) Differences: changes made in product name, price, measurement, type, alt, category, and stock fields | | | | | | | | | | | | | |
|---|-----------------|-------|-------------|--|---|---|--|---|--------------|-------|---------|----------|----------|
| id | name | price | measurement | typeName | image | contextualImageId | imageAlt | url | categoryPath | stock | created | modified | lastEdit |
| 44.99 | Kid's armchair | | | | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/PÖÄNG-Kid-s-armchair-birch-veneerMedåsog-dinosaur-pattern--https://www.ikea.com/us/en/itm/ | PÖÄNG Kid's armchair, birch veneer/Medåsog dinosaur pattern | https://www.ikea.com/us/en/ | | | | | |
| 169.00 | 26.3/8x29.7/8 | | | Armchair | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/ULRIKSBERG-Armchair-rattananthracite--https://www.ikea.com/us/en/itm/ | ULRIKSBERG Armchair, rattan/anthracite | https://www.ikea.com/us/en/ | | | | | |
| 89.00 | 14.1/8x20.7/8 | | | Cat bed | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/UTSÅLD-Cat-house-on-legs--https://www.ikea.com/us/en/itm/ | UTSÅLD Cat house on legs, rattan | https://www.ikea.com/us/en/ | | | | | |
| 765.00 | 83.1/8 x 26.5/8 | | | 3-seat conversation set, outdoor anthracite EDIT | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/REVSÅR-3-seat-conversation-set-outdoor-anthracite-dark-gray-EDIT--https://www.ikea.com/us/en/itm/ | REVSÅR 3-seat conversation set, outdoor anthracite, dark gray EDIT | https://www.ikea.com/us/en/ | | | | | |
| 16.00 | 14.1/8x20.7/8 | | | Side table, reddish brown/pink | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/TESAMMANS-Side-table--https://www.ikea.com/us/en/itm/ | TESAMMANS Side table | https://www.ikea.com/us/en/ | | | | | |
| 49.99 | 39.3/8 x 17.5/4 | | | Bench with storage, pine | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/PERJUHAN-Bench-with-storage-pine--https://www.ikea.com/us/en/itm/ | PERJUHAN Bench with storage, pine | https://www.ikea.com/us/en/ | | | | | |
| 199.99 | 50 x 60.1/4 | | | big table | https://www.ikea.com/us/en/ | https://www.ikea.com/us/en/itm/HAUGA-Big-table--https://www.ikea.com/us/en/itm/ | | | | | | | |

| | | | | | | |
|--------|------------|--------------------------------------|---|---|---|---|
| 299.00 | 22 cu ft | French door refrigerator | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | OVERSKAQ French door refrigerator, 4-door/stainless steel color, 22 cu ft | https://www.ikea.com/us/en/inr/ |
| 299.00 | 4.0 cu ft | Fridge with freezer compartment | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | LAGAN Fridge with freezer compartment, compact/white, 4.0 cu ft | https://www.ikea.com/us/en/inr/ |
| 829.00 | 13.9 cu ft | Top-freezer refrigerator | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | FRYSBAR Top-freezer refrigerator, Stainless steel, 13.9 cu ft | https://www.ikea.com/us/en/inr/ |
| 749.00 | 13.9 cu ft | Top-freezer refrigerator | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | LAGAN Top-freezer refrigerator, white, 13.9 cu ft | https://www.ikea.com/us/en/inr/ |
| 119.00 | 2 cu ft | Refrigerator | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | TILLREDA Refrigerator, freestanding/white, 2 cu ft | https://www.ikea.com/us/en/inr/ |
| 133.00 | 30x26x20 | Top cab f/ fridge/freezer w/ 2 doors | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | SEKTION Top cab f/ fridge/freezer w/ 2 doors, white/Veddinge white, 30x2 | https://www.ikea.com/us/en/inr/ |
| 185.00 | 36x24x10 | Top cab f/ fridge/freezer w/ push-op | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | SEKTION Top cab f/ fridge/freezer w/ push-op, white/Axstad matt white, 36x24x10 | https://www.ikea.com/us/en/inr/ |
| 185.00 | 36x24x10 | Top cabinet for fridge/freezer | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | SEKTION Top cabinet for fridge/freezer, white/Axstad matt white, 36x24x10 | https://www.ikea.com/us/en/inr/ |
| 289.99 | 26 x 52 | Gaming chair, Bomstad black | https://www.ikea.com/us/en/inr/ | https://www.ikea.com/us/en/inr/ | MATCHSPEL Gaming chair, Bomstad black | https://www.ikea.com/us/en/inr/ |

Showing product data in table AFTER edit (2) Differences: changes made in product name, price, measurement, type, alt, category, and stock fields

Checklist Items (1)

#1 Show a before and after screenshot of the record

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|--|--------|--|
| <input checked="" type="checkbox"/> #1 | 1 | 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 |
| <input checked="" type="checkbox"/> #2 | 1 | Briefly describe the validations for the applicable fields |

Response:

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

Briefly describe the validations for the applicable fields

On the list or view page, a product has a "edit" button associated with a specific product. When a user clicks that button it gets the (edit_product.php) url along with the product "id" - (included at the end of the url) to get that specific page.

The edit page retrieves the product id from the GET method and stores it in a variable "id". If the value of "id" was LESS than -1 then it redirects the user back to the list page and flashes a message saying it was invalid. If the "id" value was GREATER than -1, it can proceed to fetch the related product data from the db table.

- It begins making the query by starting it with the SELECT statement to select the columns (name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock) from the Products table with a WHERE - to see if the "id" matches the "id" we grabbed earlier.
- This ensures the data grabbed belongs to that specific product since every id is unique.
- After preparing the query it can execute it and fetch the product. Once it has the product details, it stores it in the variable "product".

Then, if "product" is not empty, it will begin to prepare the edit form.

- The variable "form" is assigned an array with fields for name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, and stock.

A foreach loop iterates through each record in "form" and checks if the value "name" exists in the keys of

- A foreach loop iterates through each record in "form" and checks if the value "name" exists in the keys of "product". If it does exist in "product" it means there is a related value for the form.
- It proceeds to assign that value (from "product") to the value of the field in the form. This is how the edit form gets pre-filled with the corresponding values of the product.

Whenever a user edits the form and submits it, it will trigger the "onsubmit" attribute and "validate" function. The validation function is the client-side JavaScript validation that checks the product name, price, measurement, type, image, contextual image, alt text, url, category, and stock.

- After grabbing the values using the form object and storing them in variables, it checks if they are empty by looking at the length of the variable with ".length==0".
- Using regex: For name, it checks if it contains numbers or is longer than 32 characters. For price, it also checks if there's anything other than numbers and if it's longer than 7 digits. For measurement, it checks if it contains symbols (besides /) or is longer than 75 characters. For type, it checks if it's longer than 100 characters. For image, contextual image, and url, it checks if the format of the URLs is valid. For category, it also checks if it contains symbols (other than _ and &) or if it's longer than 100 characters. For stock, it checks if there's anything other than numbers.
- The "onsubmit" takes a boolean value, so if any of these validations returns a false, then warning messages are displayed. However, if none of the validations fail, it returns true and sends the data with the POST method.

So, if the client-side validation is valid (true), the form data is sent to the server-side, and it goes through PHP validations.

- First, it checks if the POST array contains name, price, measurement, typeName, img, contextualImageUrl, alt, url, category, stock. If it does, it retrieves the data, then sets it to variables and begins validating the data.
- It proceeds to check if any of the variables are empty by using an empty() function.
- Also using the same regex as in the js validations: For name, it checks if it contains numbers or is longer than 32 characters. For price, it also checks if there's anything other than numbers and if it's longer than 7 digits. For measurement, it checks if it contains symbols (besides /) or is longer than 75 characters. For type, it checks if it's longer than 100 characters. For image, contextual image, and url, it checks if the format of the URLs is valid. For category, it also checks if it contains symbols (other than _ and &) or if it's longer than 100 characters. For stock, it checks if there's anything other than numbers.

If the field fails the validation, it returns true for "hasError" and flashes error messages to the user. However, if the field passes the validation, it returns false for "hasError" (meaning there are no errors) and proceeds.

Continuing with the edit, an if-statement checks if the field 'name' is set in the POST array.

If it is, then a foreach loop iterates through all the records in POST and checks if there are any keys not part of: name, price, measurement, type, image, contextual image, alt, url, category, stock, and removes them. This cleaned-up data is stored in the variable "quote".

Then, it begins the query by starting with UPDATE 'Products' (the table) SET. For each record in "quote", it appends the current key and its value to "query". If "params" is not empty it appends a comma to "query" to separate each key-value. After it goes through each record, a WHERE id = :id is appended to the end of the constructed query - to see if the "id" matches the "id" we grabbed earlier.

After preparing the query it can execute it and UPDATE the record with the corresponding "id". If the update works, it flashes the user with a successful updated record message.

Task #5 - Points: 1

Text: Add related links

COLLAPSE ^

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Include the heroku prod link for this page |
| #2 | 1 | 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 |

URL #1

<https://github.com/yaneliii/yc73-it202-008/pull/53>

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/54>

URL #3

<https://github.com/yaneliii/yc73-it202-008/pull/56>

URL #4

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/edit_product.php?id=1

Delete Handling (1 pt.)

[^ COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots related to delete

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Show the success message of a delete |
| #2 | 1 | Show any error messages of a failed delete (like id not being passed) |
| #3 | 1 | Show the code related to the delete processing |

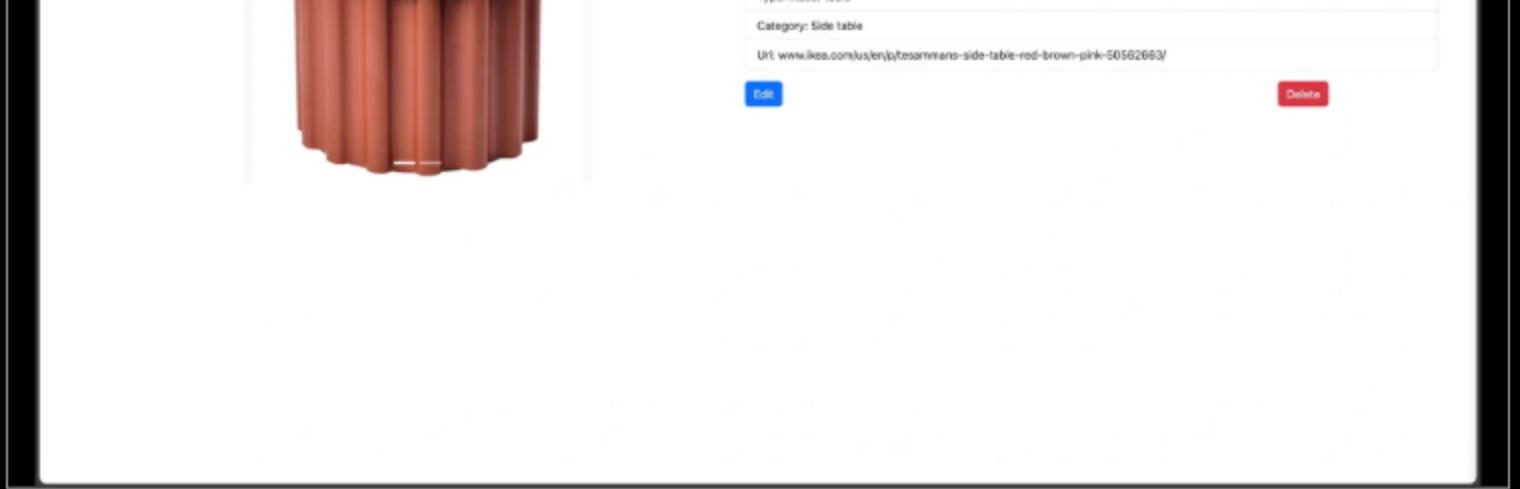
Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a web browser window with the URL https://yc73-it202-008-dev-2cc96975f868.herokuapp.com/Project/admin/view_product.php?id=380. The page title is "Product: Tablee". On the left, there is a large image of a light-colored, round-top rectangular table. To the right of the image, there is a detailed product description table with the following information:

| Tablee | |
|--------------|-----------------|
| Name: | Tablee |
| Price: | 150.00 |
| Measurement: | 14 1/8 x 20 7/8 |
| Stock: | 25 |
| Type: | Indoor table |



Showing product that will get deleted

Checklist Items (0)

| Product Name | Min Price | Max Price | Type | Category | Sort | Order | Limit |
|--------------|-----------|-----------|------|----------|---------|-------|-------|
| Name | Min Price | Max Price | Type | Category | Created | - | 50 |

Show a error messages of a failed delete (id not being passed)

Checklist Items (0)

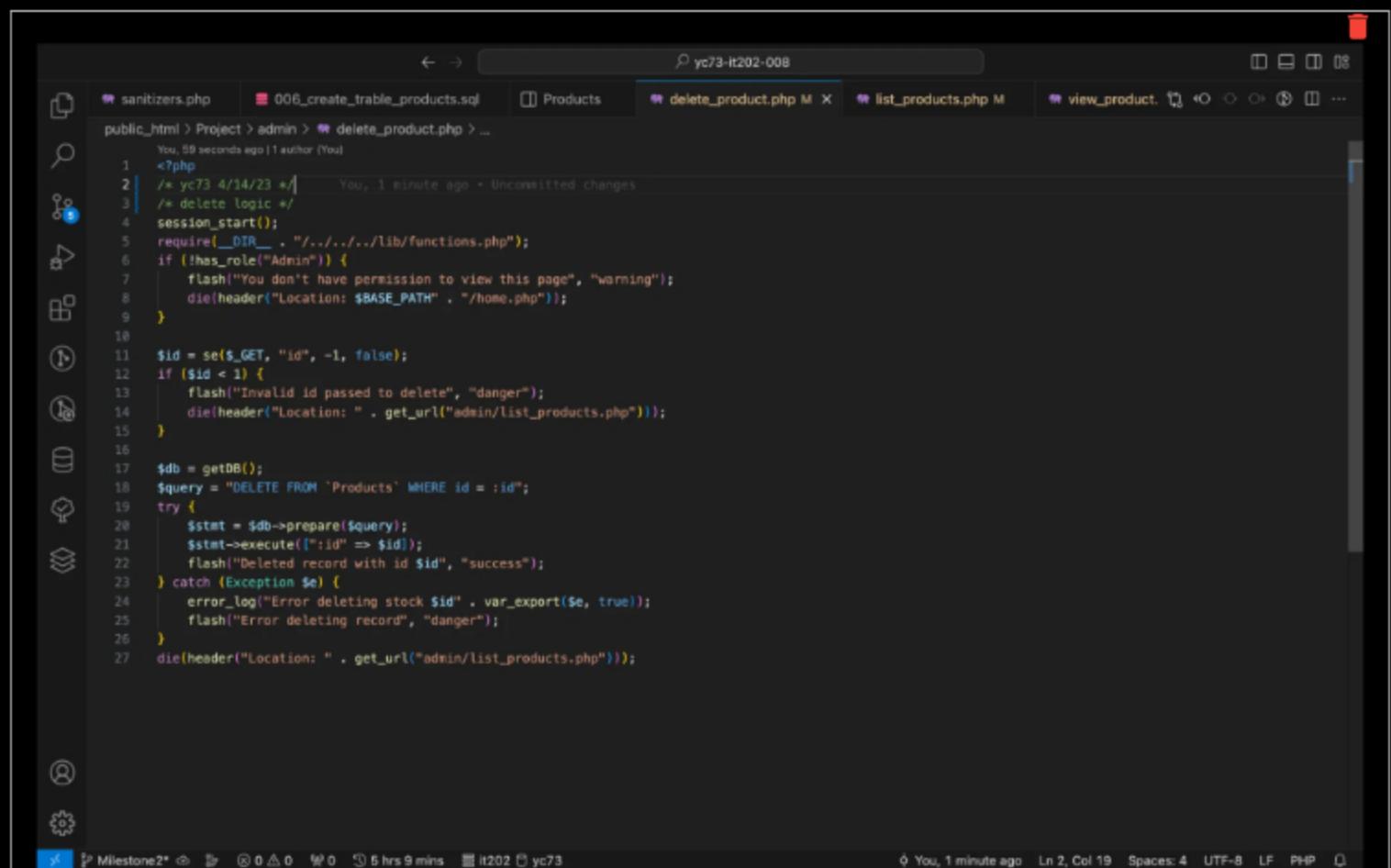
| Product Name | Min Price | Max Price | Type | Category | Sort | Order | Limit |
|--------------|-----------|-----------|------|----------|---------|-------|-------|
| Name | Min Price | Max Price | Type | Category | Created | - | 50 |

| name | price | typeName | categoryPath | stock | is_api | Actions |
|--------------|---------|------------------------------------|------------------|-------|--------|---|
| MATCHPEL | 289.99 | Gaming chair, Birsetad black | Chairs | 49 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 185.00 | Top cabinet for fridge/freezer | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 185.00 | Top cab f fridge/freezer w push-op | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SEKTION | 133.00 | Top cab f fridge/freezer w 2 doors | Kitchen cabinets | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| TILLREDA | 119.00 | Refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LAGAN | 249.00 | Top-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FRYSBAR | 829.00 | Top-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| LAGAN | 299.00 | Fridge with freezer compartment | Kitchen systems | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| ÖVERSÄDÅG | 1999.00 | French door refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| VÄLGRUNDAD | 1299.00 | Bottom-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| SUPERKALL | 1399.00 | Built-in, panel-ready refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| FÄRSKHET | 1199.00 | Bottom-freezer refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| STJÄRNSTATUS | 2699.00 | French door refrigerator | Appliances | 0 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |
| HAUGA | 179.00 | High cabinet with 2 doors, white | Cabinet & shelf | 12 | 1 | <button>View</button> <button>Edit</button> <button>Delete</button> |

Show a success message of a delete

Checklist Items (1)

#1 Show the success message of a delete



The screenshot shows a code editor with several tabs open. The active tab is 'delete_product.php' containing the following PHP code:

```

public_html > Project > admin > delete_product.php > ...
You, 59 seconds ago | 1 author (You)
<?php
/* yc73 4/14/23 */
/* delete logic */
session_start();
require(__DIR__ . "/../../../../lib/functions.php");
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: $BASE_PATH" . "/home.php"));
}
$id = isset($_GET, "id", -1, false);
if ($id < 1) {
    flash("Invalid id passed to delete", "danger");
    die(header("Location: " . get_url("admin/list_products.php")));
}
$db = getDB();
$query = "DELETE FROM `Products` WHERE id = :id";
try {
    $stmt = $db->prepare($query);
    $stmt->execute([":id" => $id]);
    flash("Deleted record with id $id", "success");
} catch (Exception $e) {
    error_log("Error deleting stock $id" . var_export($e, true));
    flash("Error deleting record", "danger");
}
die(header("Location: " . get_url("admin/list_products.php")));

```

Showing the code related to the delete processing

Checklist Items (1)

#3 Show the code related to the delete processing

Task #2 - Points: 1

Text: Screenshots of the data

[COLLAPSE ▾](#)

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Show a before and after screenshot of the DB data |
| #2 | 1 | Note what changed (i.e., record removed or soft delete value changed) |
| #3 | 1 | Clearly caption screenshots |

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

| | id | api_id | name | price | measurement | typeName | image | contextualImage |
|------|-----|-----------|---------------|---------|-----------------|----------------------------------|---|---|
| > 47 | 302 | s90570883 | UTSAAD | 69.00 | 14 1/8x20 7/8 | Cat bed | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 48 | 314 | 69544631 | REVSKÄRRRRRRR | 755.00 | 83 1/8 x 25 5/8 | 3-seat conversation set, outdo | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 49 | 317 | 21459483 | TESAMMANS | 16.00 | 14 1/8x20 7/8 | Side table, redbrown/pink | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 50 | 376 | 60485339 | PERJODHAN | 49.99 | 39 3/8 x 17 3/4 | Bench with storage, pine | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 51 | 377 | 1234567 | Desk | 199.99 | 50 x 60 1/4 | big table | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 52 | 378 | 20415046 | HAUGA | 179.00 | 27 1/2 x 78 3/8 | High cabinet with 2 doors, whi | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 53 | 380 | 98765432 | Tablee | 130.00 | 14 1/8 x 20 7/8 | indoor table | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 54 | 381 | 20522287 | STJÄRNSTATUS | 2699.00 | 22.6 cu.ft | French door refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 55 | 382 | 30462156 | FÄRSKHET | 1199.00 | 10.4 cu.ft | Bottom-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 56 | 383 | 10462156 | SUPERKALL | 1399.00 | 8.8 cu.ft | Built-in, panel-ready refrigerat | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 57 | 384 | 70462158 | VÄLGRUNDAD | 1299.00 | 19 cu.ft | Bottom-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 58 | 385 | 60462154 | ÖVERSÅKLIG | 2999.00 | 22 cu.ft | French door refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 59 | 386 | 50490119 | LAGAN | 299.00 | 4.0 cu.ft | Fridge with freezer compartm | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 60 | 387 | 80462148 | FRYSBAR | 829.00 | 13.9 cu.ft | Top-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 61 | 388 | 00462152 | LAGAN | 749.00 | 13.9 cu.ft | Top-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 62 | 389 | 20497023 | TILLREDA | 119.00 | 2 cu.ft | Refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |

Showing the date in the DB table BEFORE deletion

Checklist Items (0)

| | id | api_id | name | price | measurement | typeName | image | contextualImage |
|------|-----|-----------|---------------|--------|-----------------|--------------------------------|---|---|
| > 47 | 302 | s90570883 | UTSAAD | 69.00 | 14 1/8x20 7/8 | Cat bed | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |
| > 48 | 314 | 69544631 | REVSKÄRRRRRRR | 755.00 | 83 1/8 x 25 5/8 | 3-seat conversation set, outdo | https://www.ikea.com/us/en/fir | https://www.ikea.com/us/en/fir |

| | | | | | | | | |
|------|-----|-----------|--------------|---------|-----------------|----------------------------------|---|---|
| > 49 | 317 | 21459483 | TESAMMANS | 16.00 | 14 1/8x20 7/8 | Side table, redbrown/pink | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 50 | 376 | 60485339 | PERJOHAN | 49.99 | 39 3/8 x 17 3/4 | Bench with storage, pine | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 51 | 377 | 1234567 | Desk | 199.99 | 50 x 60 1/4 | big table | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 52 | 378 | 20415046 | HAUGA | 179.00 | 27 1/2 x 78 3/8 | High cabinet with 2 doors, whi | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 53 | 381 | 20522287 | STJÄRNSTATUS | 2699.00 | 22.6 cu.ft | French door refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 54 | 382 | 30462155 | FÄRSKHET | 1199.00 | 10.4 cu.ft | Bottom-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 55 | 383 | 10462156 | SUPERKALL | 1399.00 | 8.8 cu.ft | Built-in, panel-ready refrigerat | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 56 | 384 | 70462158 | VÄLGRUNDAD | 1299.00 | 19 cu.ft | Bottom-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 57 | 385 | 60462154 | ÖVERSÅDLIG | 2999.00 | 22 cu.ft | French door refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 58 | 386 | 50490119 | LAGAN | 299.00 | 4.0 cu.ft | Fridge with freezer compartment | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 59 | 387 | 80462148 | FRYSBAR | 829.00 | 13.9 cu.ft | Top-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 60 | 388 | 00462152 | LAGAN | 749.00 | 13.9 cu.ft | Top-freezer refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 61 | 389 | 20497023 | TILLREDA | 119.00 | 2 cu.ft | Refrigerator | https://www.ikea.com/us/en/fir | https://www.ikea.com |
| > 62 | 390 | s09462874 | SEKTION | 133.00 | 30x24x20 | Top cab f fridge/freezer w 2 dr | https://www.ikea.com/us/en/fir | https://www.ikea.com |

Milestone2* ⏪ 0 0 △ 0 ⏴ 0 0 ④ 4 hrs 53 mins ■ It202 ☰ yc73

Q

Showing the date removed in the DB table AFTER deletion Difference: the record (with id 378) was deleted
(irrecoverable; was removed with a hard delete)

Checklist Items (2)

#1 Show a before and after screenshot of the DB data

#2 Note what changed (i.e., record removed or soft delete value changed)

Task #3 - Points: 1

Text: Explain the delete logic

Checklist

*The checkboxes are for your own tracking

| # | Points | Details |
|----|--------|---|
| #1 | 1 | Is it a soft or hard delete |
| #2 | 1 | Are there any necessary roles or restrictions? (can only delete their data, can only be done by admin, etc) |
| #3 | 1 | Provide the high-level steps for handling the DB lookup and handling the delete |

Response:

First, the code checks if the user trying to delete the product is an admin. Only admins can delete products, we don't want regular users (non-admins) to be able to delete products. If they are NOT an admin they are redirected to the home page. If they are, then it proceeds.

To delete a product, each product had "delete" button (on the list or view page) that is associated with a specific product. When a user clicks that button it gets the (delete_product.php) url along with the product "id" - (included at the end of the url) to get that specific product.

When clicked, it retrieves the product id from the GET method and stores it in a variable "id". If the value of "id" was LESS than 1 then it redirects the user back to the list page and flashes a message saying it was invalid. If the "id" value was GREATER than 1, it can proceed to create the query.

It begins the query with a DELETE statement FROM the table Products along with a WHERE id = :id - to see if

- the "id" matches the "id" we grabbed earlier.

Being that a DELETE statement is used, this will be a hard delete because after it deletes the existing record from the db table it cannot be recovered.

After preparing the query it can execute it and delete the record with the corresponding "id". If the delete works, it flashes the user with a successful deleted record message.

Task #4 - Points: 1

Text: Add the pull request link for the branch related to this feature

Details:

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

URL #1

<https://github.com/yaneliii/yc73-it202-008/pull/54>

Misc (1 pt.)

[^ COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshot of your project board from GitHub (tasks should be in the proper column)

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a GitHub project board titled 'Yanelli's IT202 Project'. It has three columns: 'Todo', 'In Progress', and 'Done'. The 'Todo' column contains one item: 'This item hasn't been started'. The 'In Progress' column contains one item: 'This is actively being worked on'. The 'Done' column contains several items, each with a small circular icon and some text. The items are:

- MS1 - Any output messages/errors should be "user friendly"
- MS1 - User will be able to see their profile
- MS1 - User will be able to edit their profile
- MS2 - Define the appropriate table or tables for your API
- MS2 - Data Creation Page
- MS2 - Data List Page (many items)
- MS2 - View Data Details Page (single item)
- MS2 - Edit Data Details Page (single item)

MS2 - Edit Data Page
yc73-lt202-008 #40
MS2 - Delete Handling
yc73-lt202-008 #50
MS2 - API Handling
yc73-lt202-008 #50

+ Add Item + Add Item + Add Item

Showing project board from GitHub

Task #2 - Points: 1

Text: Provide a direct link to the project board on GitHub

URL #1

<https://github.com/users/yaneliii/projects/3>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

I had a lot of issues attempting to manually create data and fetch data from my API. At first it was issues with my insert logic, I learned how to make it take in multiple inserts. I also had issues with what post data the fetch and create wanted - (fetch wanted the "keyword" while create wanted something related to its form like "api_id"). These issues allowed me to understand the flow and logic of how the data was being moved around as well as how the data is received from the api or from a form and how to manipulate it (to do things like insertions, pre-fill forms, or view specific products based on its id).

Task #4 - Points: 1

Text: WakaTime Screenshot

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: Large View

Small

Medium

Large

WakaTime

Upgrade

https://wakatime.com/projects/yc73-lt202-008?start=2024-04-13&end=2024-04-19

Files

- 20 hrs 8 mins ..ct/admin/create_product.php
- 5 hrs 27 mins ..ject/admin/edit_product.php
- 4 hrs 40 mins ..action_index/Products/selected.php

Branches

- 26 hrs 15 mins MS2-API-Integration-Admin
- 8 hrs 45 mins MS2-List-Products-Enhancement
- 8 hrs 16 mins MS2-Form-Validation

Dashboard
+ Insights
@ Teams
Projects
Invoices
Goals
Shareables
Leaderboards
Integrations
Supported IDEs

Bug
API Docs
Pricing
Community
FAQ
Plugin Status
About

| | | | |
|---------------|----------------------------------|---------------|---------------------------|
| 4 hrs 40 mins | public_html/Project/api/resource | 2 hrs 20 mins | Milestone2 |
| 3 hrs 28 mins | _ect/admin/list_products.php | 22 mins | MS2-API-Integration |
| 1 hr 29 mins | _ject/admin/view_product.php | 7 mins | MS2-Form-Fetch-Validation |
| 1 hr 12 mins | lib/store_api.php | 6 mins | MS2-DB-Helper |
| 53 mins | _ect/admin/practice_list.php | 3 secs | Unknown |
| 37 mins | _ic_html/Project/profile.php | | |
| 37 mins | _ct/admin/delete_product.php | | |
| 35 mins | __create_table_products.sql | | |
| 25 mins | partials/nav.php | | |
| 20 mins | lib/sanitizers.php | | |
| 20 mins | lib/functions.php | | |
| 18 mins | lib/db_helpers.php | | |
| 14 mins | _c_html/Project/register.php | | |
| 14 mins | lib/err | | |
| 12 mins | partials/input_field.php | | |
| 11 mins | _ql/product_table_sample.sql | | |
| 6 mins | partials/table.php | | |
| 6 mins | lib/load_api_keys.php | | |
| 5 mins | _r_table_products_ic_api.sql | | |
| 4 mins | _ect/admin/view_product.php | | |
| 4 mins | __create_table_userroles.sql | | |
| 4 mins | lib/session_store.php | | |
| 3 mins | public_html/Project/login.php | | |
| 2 mins | _ml/Project/products_api.php | | |
| 2 mins | _roject/admin/list_roles.php | | |
| 1 min | lib/api_helper.php | | |
| 1 min | lib/render_functions.php | | |
| 58 secs | _ql/create_table_sample.sql | | |
| 39 secs | lib/db_helpers | | |
| 17 secs | _tm/Project/sql/init_db.php | | |
| 14 secs | lib/get_url.php | | |
| 6 secs | lib/db_helper.php | | |
| 3 secs | Untitled-1 | | |
| 2 secs | _object/admin/create_role.php | | |
| 0 secs | _ject/admin/assign_roles.php | | |

Showing WakaTime

End of Assignment