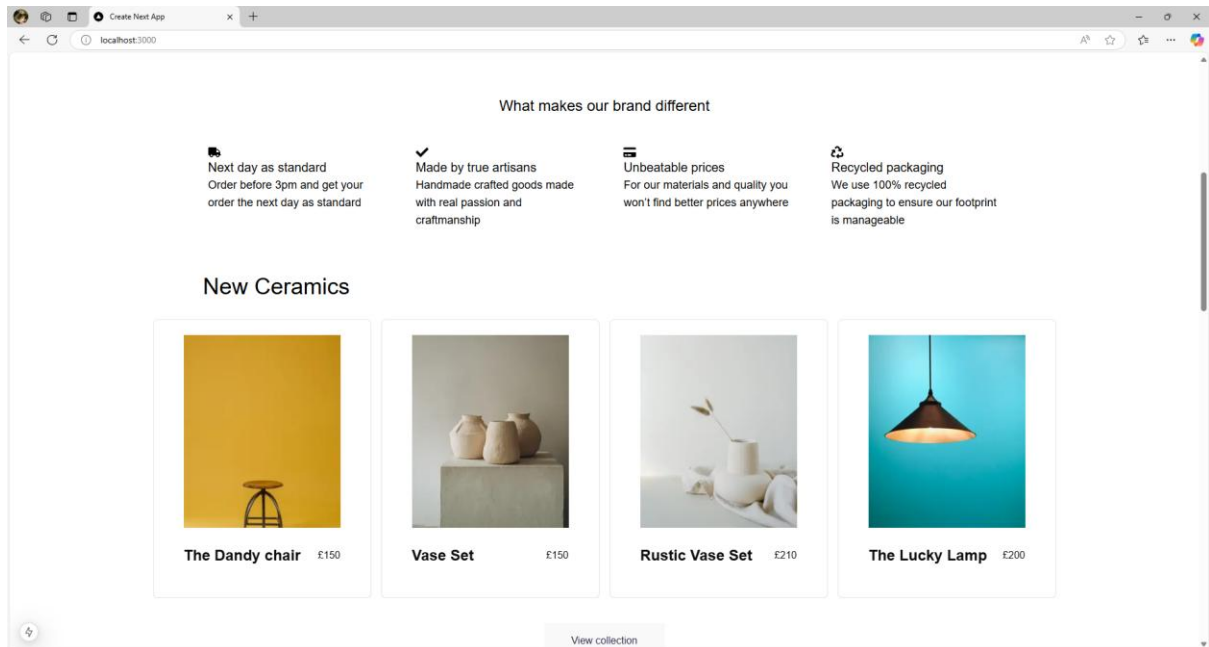


# "Day 4 - Dynamic Frontend Components - [Muniba e-shop]"

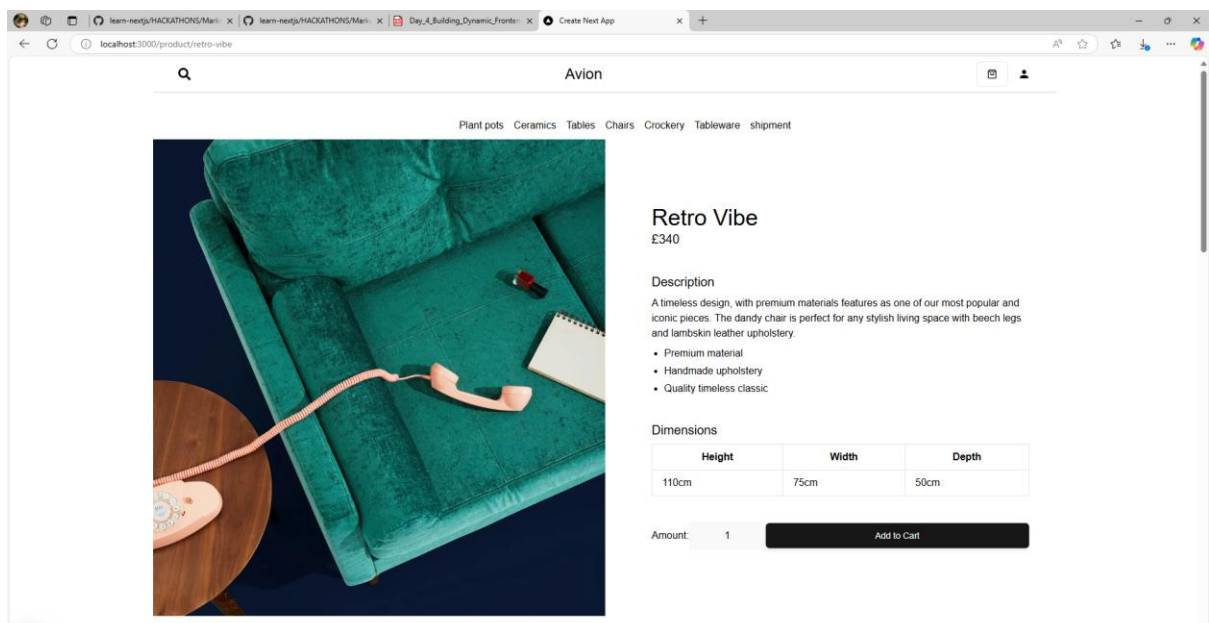
## Product Listing Component:

Product List with filter of New ceramic tag.



## Product Detail Page:

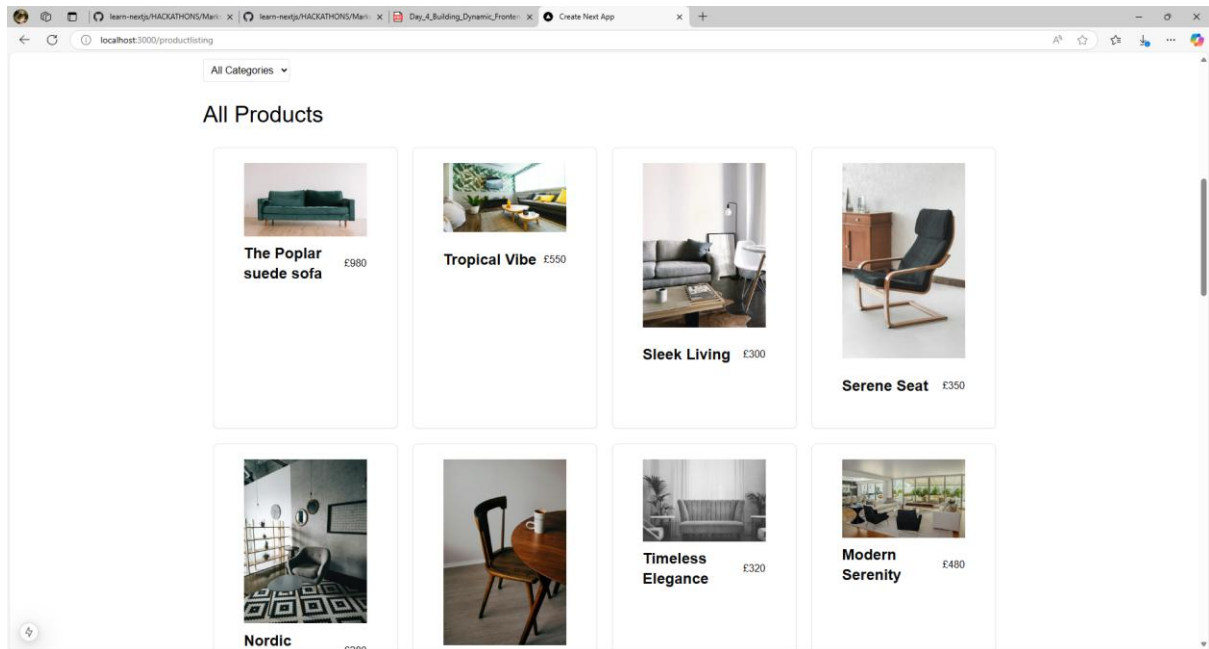
A Dynamic product detail page that is fetching data from sanity on the base of slug field of products.



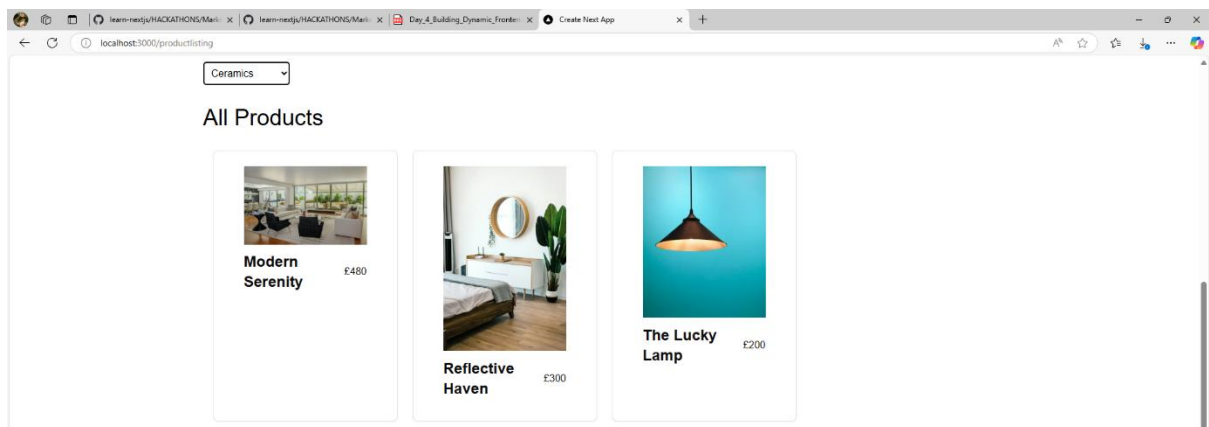
## Category Component:

Category component that has a dropdown list to filter categories coming from sanity.

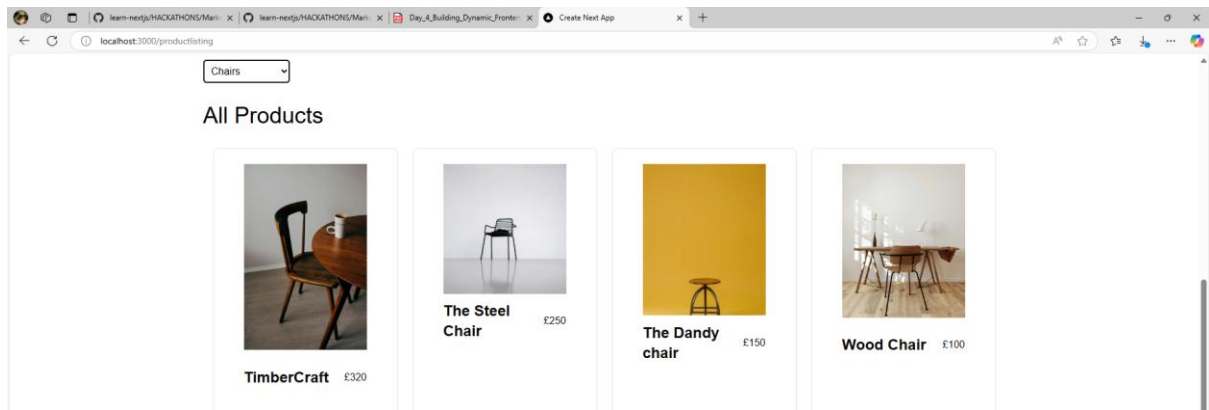
1- All products:



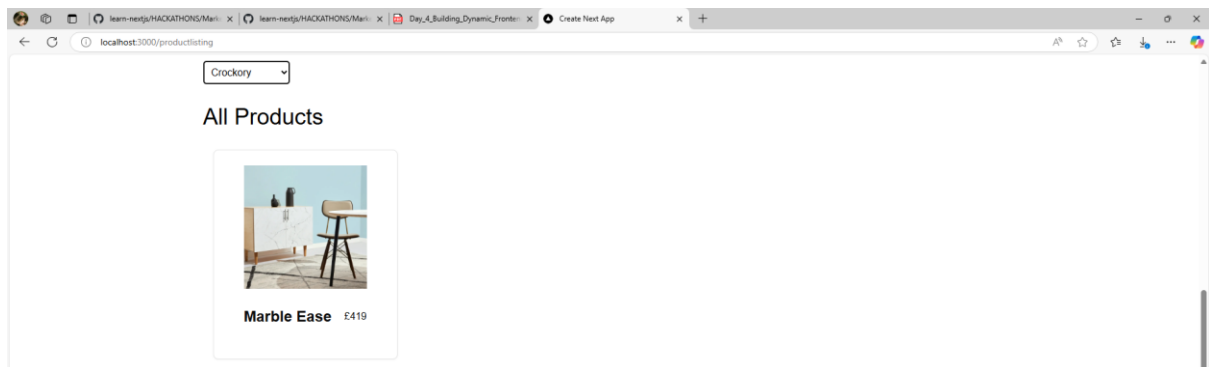
2- Ceramics



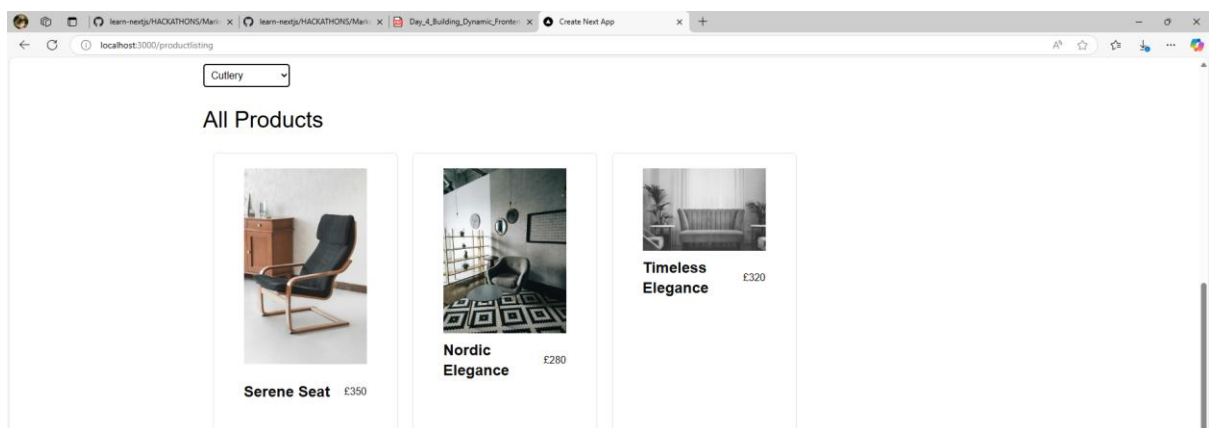
3- Chairs



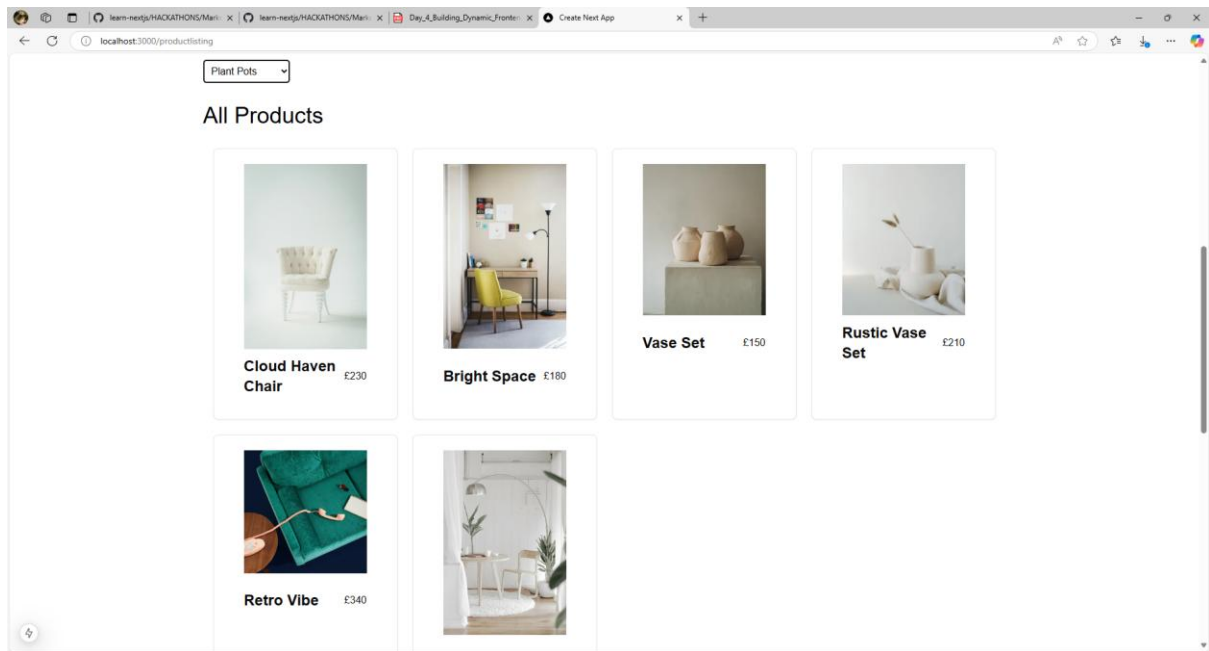
#### 4- Crockery



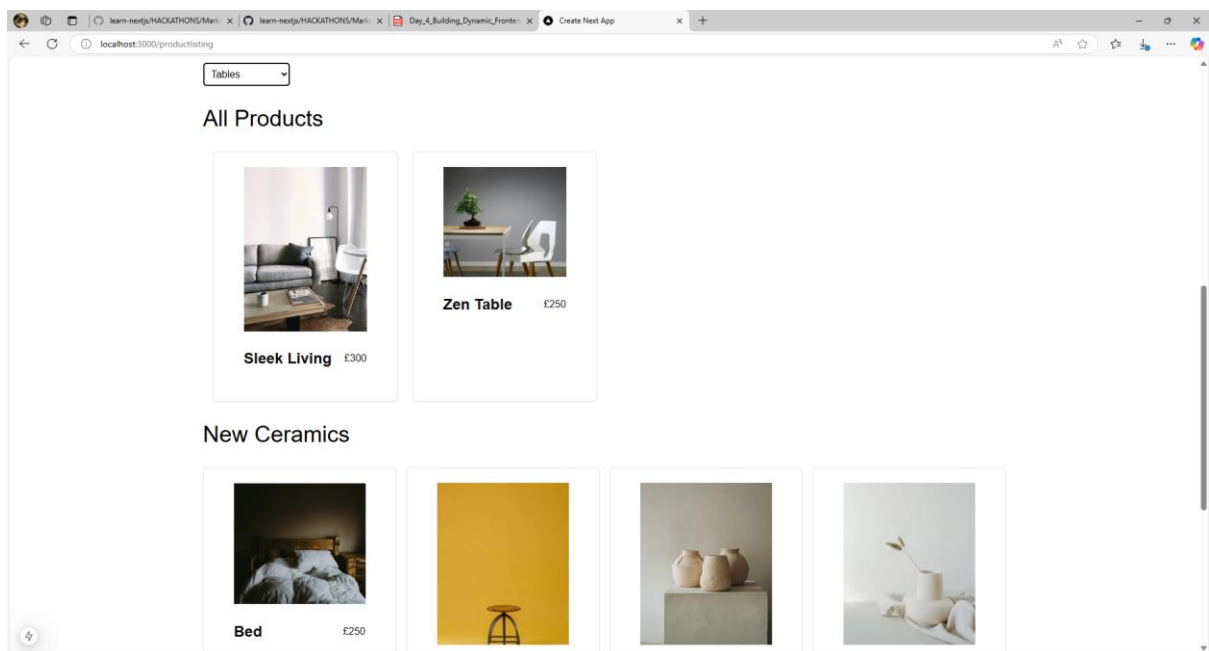
#### 5- Cutlery



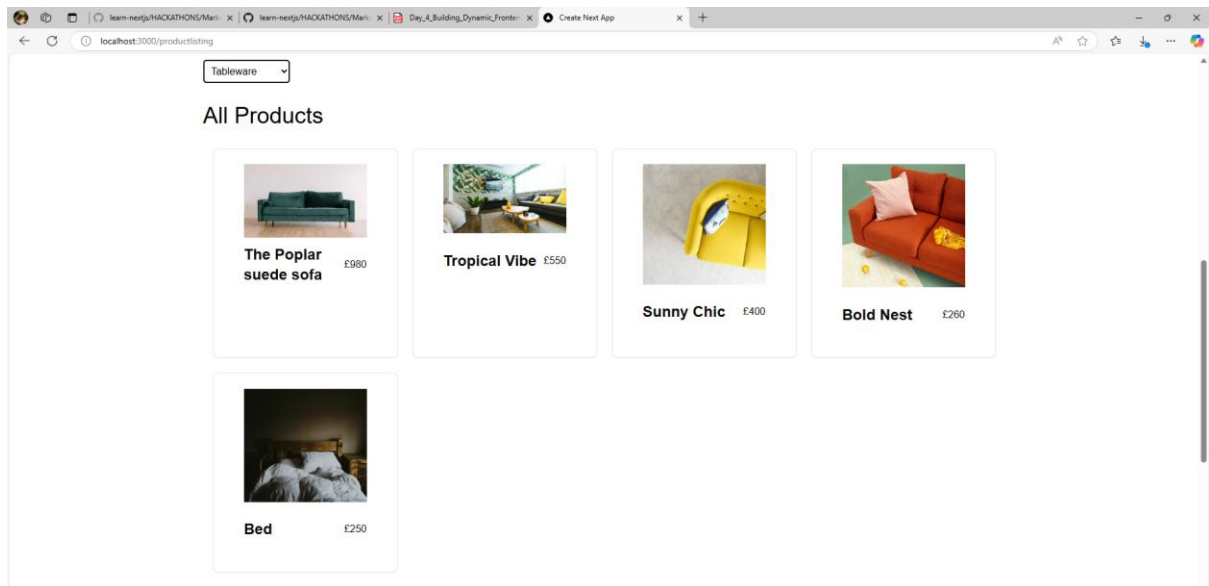
#### 6- Plant pots



## 7- Tables

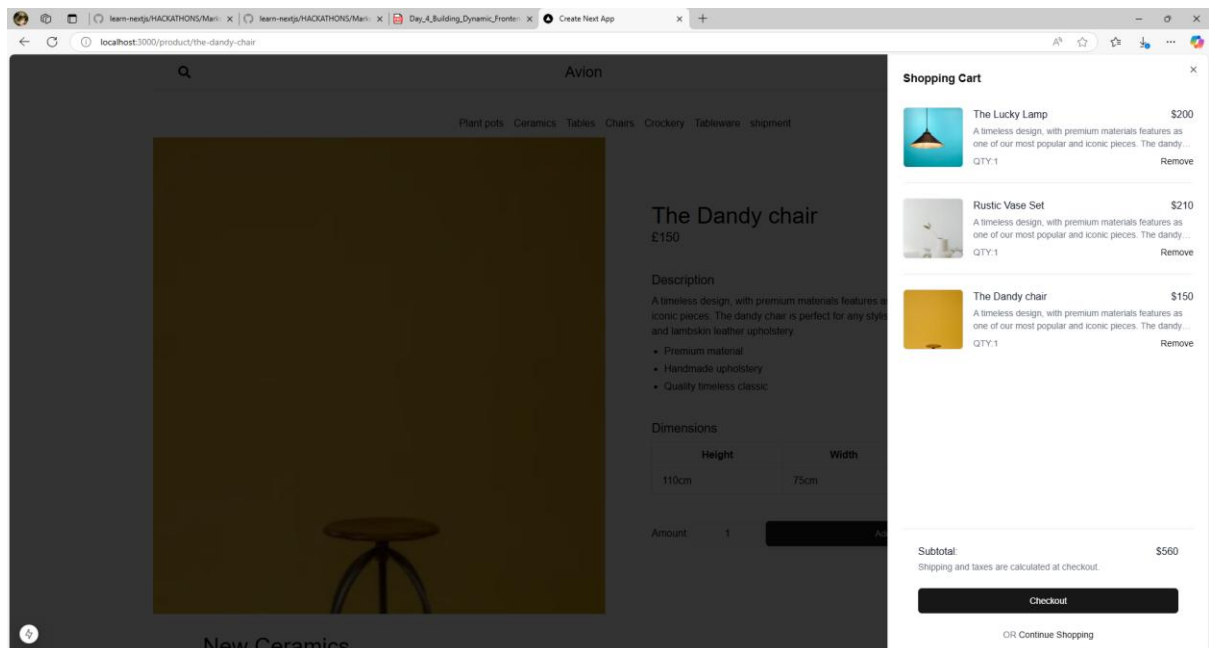


## 8- Tableware



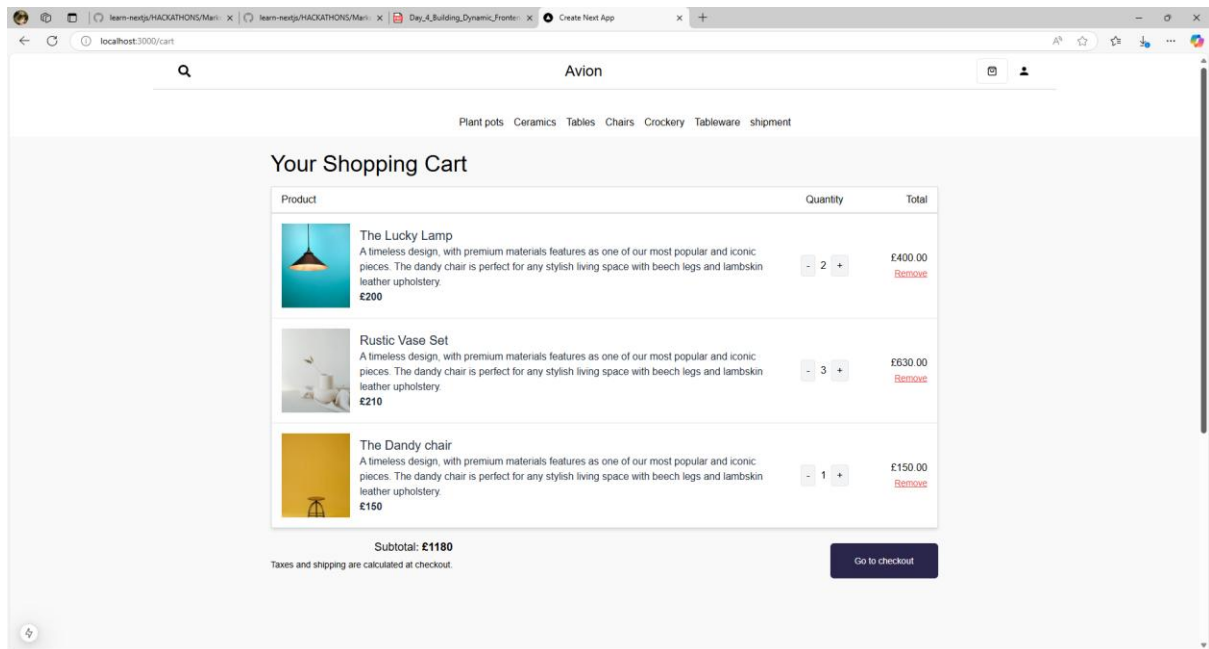
## Add to Cart Component:

ShoppingCartModal Component display on right side when click add to cart or cart button on navbar design using shadcn ui sheet and shopping cart hook.



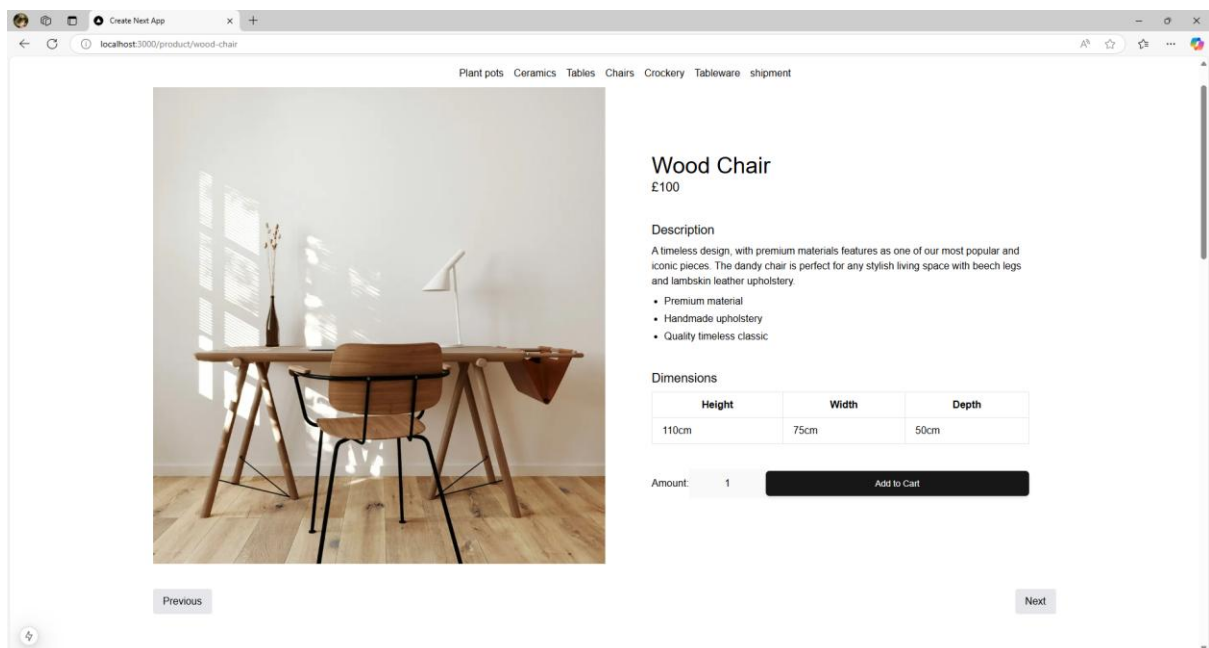
## Cart Page:-

For add or subtract quantities of products that add in cart and show price of total products then subtotal of all products.



## Pagination:

A pagination from shadcn ui added to my product detail dynamic route so we can check the detail of previous and next product through pagination:



## Code Snippet for important components and pages:

### 1- Product listing with new ceramic tag:

```

1 // Code hereimport client from "@sanity/lib/client";
2 import Image from "next/image";
3 import { Carousel, CarouselContent, CarouselItem } from "./ui/carousel";
4 import { Card, CardContent, CardFooter, CardHeader } from "./ui/card";
5 import Link from "next/link";
6
7 interface Product {
8   image: string;
9   name: string;
10  price: number;
11  description: string;
12  slug: { current: string }; // Correct slug structure from Sanity
13  price_id: string;
14  tags: string[];
15 }
16
17 const Ceramics = async () => {
18   // Sanity query to fetch products with the 'new ceramics' tag
19   const query = `*[ _type == 'product' && 'new ceramics' in tags ]{
20     "image": image.asset->url,
21     name,
22     price,
23     description,
24     slug, // Fetch slug object
25     price_id,
26     tags
27   }`;
28
29   const res: Product[] = await client.fetch(query);
30
31   return (
32     <div className="md:max-w-[1440px] w-full h-auto mx-auto">
33       { /* New Ceramics Section */ }
34       <div>
35         <div className="w-[1280px] mx-auto">
36           <h1 className="text-4xl my-8">New Ceramics</h1>
37         </div>
38
39         { /* Carousel Container */ }
40         <Carousel className="w-full max-w-full">
41           <CarouselContent>
42             { /* Map through the products */ }
43             {res.map((product: Product, index: number) => (
44               <CarouselItem
45                 key={index}
46                 className="md:basis-1/2 lg:basis-1/4"
47               >
48                 <Card className="w-full">
49                   <CardContent className="flex flex-col items-center justify-center">
50                     <CardHeader>
51                       <Link href={` /product/${product.slug.current} `}>
52                         <Image
53                           src={product.image || "/product1.png"}
54                           alt={product.name}
55                           height={1000}
56                           width={1000}
57                           className="h-full w-full object-cover cursor-pointer"
58                         />
59                       </Link>
60                     </CardHeader>
61                     <CardFooter className="flex items-center justify-between w-full h-[63px]">
62                       <p className="font-bold text-2xl">{product.name}</p>
63                       <p>£{product.price}</p>
64                     </CardFooter>
65                   </CardContent>
66                 </Card>
67               </CarouselItem>
68             )))
69           </CarouselContent>
70         </Carousel>
71       </div>
72     </div>
73   );
74 };
75
76 export default Ceramics;
77

```

## **2- Product detail dynamic route with pagination:**



```

100
101
102 // Import client from "@firebase/firebase-tools";
103 // Import Image from "react-image";
104 // Import Advertiser from "@app/components/advertiser";
105 // Import { metadata } from "react-adsense";
106 //
107 // Import Credentials from "@app/components/credentials";
108 // Import Results from "@app/components/results";
109 // Import Brand from "@app/components/brand";
110 //
111 //
112 // Interface Product {
113 //   price_id: string;
114 //   image: string;
115 //   name: string;
116 //   price: number;
117 //   description: string;
118 //   slug: string;
119 //   dimensions: {
120 //     height: string;
121 //     width: string;
122 //     depth: string;
123 //   };
124 // };
125 //
126 // const ProductDetails = async ({ params }: { params: Promise<slug:string> }) => {
127 //   const { slug } = await params;
128 //
129 //   // Query to fetch the current product details
130 //   const query = `? type = 'product' & slug.current = $slug | order(shop.current asc)|0 "slug", slug.current `;
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 //   //
174 //   //
175 //   //
176 //   //
177 //   //
178 //   //
179 //   //
180 //   //
181 //   //
182 //   //
183 //   //
184 //   //
185 //   //
186 //   //
187 //   //
188 //   //
189 //   //
190 //   //
191 //   //
192 //   //
193 //   //
194 //   //
195 //   //
196 //   //
197 //   //
198 //   //
199 //   //
200 //   //
201 //   //
202 //   //
203 //   //
204 //   //
205 //   //
206 //   //
207 //   //
208 //   //
209 //   //
210 //   //
211 //   //
212 //   //
213 //   //
214 //   //
215 //   //
216 //   //
217 //   //
218 //   //
219 //   //
220 //   //
221 //   //
222 //   //
223 //   //
224 //   //
225 //   //
226 //   //
227 //   //
228 //   //
229 //   //
230 //   //
231 //   //
232 //   //
233 //   //
234 //   //
235 //   //
236 //   //
237 //   //
238 //   //
239 //   //
240 //   //
241 //   //
242 //   //
243 //   //
244 //   //
245 //   //
246 //   //
247 //   //
248 //   //
249 //   //
250 //   //
251 //   //
252 //   //
253 //   //
254 //   //
255 //   //
256 //   //
257 //   //
258 //   //
259 //   //
260 //   //
261 //   //
262 //   //
263 //   //
264 //   //
265 //   //
266 //   //
267 //   //
268 //   //
269 //   //
270 //   //
271 //   //
272 //   //
273 //   //
274 //   //
275 //   //
276 //   //
277 //   //
278 //   //
279 //   //
280 //   //
281 //   //
282 //   //
283 //   //
284 //   //
285 //   //
286 //   //
287 //   //
288 //   //
289 //   //
290 //   //
291 //   //
292 //   //
293 //   //
294 //   //
295 //   //
296 //   //
297 //   //
298 //   //
299 //   //
300 //   //
301 //   //
302 //   //
303 //   //
304 //   //
305 //   //
306 //   //
307 //   //
308 //   //
309 //   //
310 //   //
311 //   //
312 //   //
313 //   //
314 //   //
315 //   //
316 //   //
317 //   //
318 //   //
319 //   //
320 //   //
321 //   //
322 //   //
323 //   //
324 //   //
325 //   //
326 //   //
327 //   //
328 //   //
329 //   //
330 //   //
331 //   //
332 //   //
333 //   //
334 //   //
335 //   //
336 //   //
337 //   //
338 //   //
339 //   //
340 //   //
341 //   //
342 //   //
343 //   //
344 //   //
345 //   //
346 //   //
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 //   //
66
```

**3- Category component that filters category:**

[illegible]

## Conclusion:

- 1- Here is the outcome of front-end ui that is rendered from the headless CMS i.e: sanity. I write groq for fetching products, product details , category and tags in different component and pages.
- 2- Challenges I have faced when filtering category on the base of slug then i filter with name in place of slug.
- 3- Secondly I have a challenge in implementing pagination in dynamic route of product detail.
- 4- At the end the outcome achieve from the 4th day is product listing, product detail dynamic route, category component and add to cart component and page.