

Nama: Idham Cholid Junarto

NPM: 22312133

```
1  "use client";
2
3  import Link from "next/link";
4
5  import { useState, useEffect } from "react";
6
7  const Sidebar = () => {
8    const [location, setLocation] = useState("");
9
10   useEffect(() => {
11     setLocation(window.location.pathname);
12   }, [location]);
13
14   return (
15     <div className="bg-gray-800 w-64 p-5 text-white hidden lg:block">
16       <h2 className="text-2xl font-bold text-center mb-8">Admin</h2>
17
18       <ul>
19         <li className="mb-5">
20           <Link
21             className={`
22               ${
23                 location !== "/" ? "" : "bg-gray-600 "
24               }block text-lg p-2 rounded hover:bg-gray-400
25             `}
26             href="/"
27           >
28             Home
29           </Link>
30         </li>
31
32         <li className="mb-5">
33           <Link
34             className={`
35               ${
36                 location !== "/admin-dashboard/laptop" ? "" : "bg-gray-600 "
37               }block text-lg p-2 rounded hover:bg-gray-400
38             `}
39             href="/admin-dashboard/laptop"
40           >
41             Laptop
42           </Link>
43         </li>
44       </ul>
45     </div>
46   );
47 };
48
49 export default Sidebar;
50
```

Fitur:

Menambahkan komponen sidebar pada halaman admin-dashboard untuk navigasi berpindah halaman atau sign-out.

```

1  "use client";
2
3  import { useRouter } from "next/navigation";
4
5  const Navbar = () => {
6    const router = useRouter();
7
8    return (
9      <header className="bg-white p-4 shadow-m">
10        <div className="flex flex-wrap gap-10 justify-between items-center">
11          <h1 className="text-3xl font-bold text-center">Admin Dashboard</h1>
12
13          <div className="flex flex-wrap items-center space-x-4">
14            <button
15              className="text-white text-lg py-1 px-2 rounded-md mx-20 hover:font-medium bg-gray-600 hover:bg-gray-900"
16              onClick={async (e) => {
17                e.preventDefault();
18
19                try {
20                  const response = await fetch("/api/sign-out", {
21                    method: "POST",
22                  });
23
24                  const { messageResponse } = await response.json();
25
26                  alert(messageResponse);
27
28                  router.refresh();
29                } catch (error) {
30                  console.log(error);
31                }
32              }}
33            >
34              Logout
35            </button>
36          </div>
37        </div>
38      </header>
39    );
40  };
41
42  export default Navbar;
43

```

Fitur:

Menambahkan komponen navbar pada halaman admin-dashboard untuk navigasi berpindah halaman atau sign-out.

Menambahkan tampilan halaman web admin-dashboard untuk menampilkan seluruh data laptop yang tersedia.

```

1  # Test client(s)
2
3  # Create a connection to the PostgreSQL server
4  conn = psycopg2.connect(
5      host="localhost",
6      port=5432,
7      user="postgres",
8      password="postgres",
9      database="postgres"
10 )
11
12 # Create a cursor object
13 cur = conn.cursor()
14
15 # Create a table
16 cur.execute("""
17     CREATE TABLE IF NOT EXISTS test_table (
18         id SERIAL PRIMARY KEY,
19         name VARCHAR(255) NOT NULL,
20         age INTEGER
21     );
22 """)
23
24 # Commit the changes
25 conn.commit()
26
27 # Insert data into the table
28 cur.execute("""
29     INSERT INTO test_table (name, age)
30     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
31 """)
32
33 # Commit the changes
34 conn.commit()
35
36 # Query the data
37 cur.execute("""
38     SELECT * FROM test_table;
39 """)
40
41 # Fetch all rows
42 rows = cur.fetchall()
43
44 # Print the results
45 for row in rows:
46     print(row)
47
48 # Close the cursor and connection
49 cur.close()
50 conn.close()
51
52 # Test client(s)
53
54 # Create a connection to the PostgreSQL server
55 conn = psycopg2.connect(
56     host="localhost",
57     port=5432,
58     user="postgres",
59     password="postgres",
60     database="postgres"
61 )
62
63 # Create a cursor object
64 cur = conn.cursor()
65
66 # Create a table
67 cur.execute("""
68     CREATE TABLE IF NOT EXISTS test_table (
69         id SERIAL PRIMARY KEY,
70         name VARCHAR(255) NOT NULL,
71         age INTEGER
72     );
73 """)
74
75 # Commit the changes
76 conn.commit()
77
78 # Insert data into the table
79 cur.execute("""
80     INSERT INTO test_table (name, age)
81     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
82 """)
83
84 # Commit the changes
85 conn.commit()
86
87 # Query the data
88 cur.execute("""
89     SELECT * FROM test_table;
90 """)
91
92 # Fetch all rows
93 rows = cur.fetchall()
94
95 # Print the results
96 for row in rows:
97     print(row)
98
99 # Close the cursor and connection
100 cur.close()
101 conn.close()
102
103 # Test client(s)
104
105 # Create a connection to the PostgreSQL server
106 conn = psycopg2.connect(
107     host="localhost",
108     port=5432,
109     user="postgres",
110     password="postgres",
111     database="postgres"
112 )
113
114 # Create a cursor object
115 cur = conn.cursor()
116
117 # Create a table
118 cur.execute("""
119     CREATE TABLE IF NOT EXISTS test_table (
120         id SERIAL PRIMARY KEY,
121         name VARCHAR(255) NOT NULL,
122         age INTEGER
123     );
124 """)
125
126 # Commit the changes
127 conn.commit()
128
129 # Insert data into the table
130 cur.execute("""
131     INSERT INTO test_table (name, age)
132     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
133 """)
134
135 # Commit the changes
136 conn.commit()
137
138 # Query the data
139 cur.execute("""
140     SELECT * FROM test_table;
141 """)
142
143 # Fetch all rows
144 rows = cur.fetchall()
145
146 # Print the results
147 for row in rows:
148     print(row)
149
150 # Close the cursor and connection
151 cur.close()
152 conn.close()
153
154 # Test client(s)
155
156 # Create a connection to the PostgreSQL server
157 conn = psycopg2.connect(
158     host="localhost",
159     port=5432,
160     user="postgres",
161     password="postgres",
162     database="postgres"
163 )
164
165 # Create a cursor object
166 cur = conn.cursor()
167
168 # Create a table
169 cur.execute("""
170     CREATE TABLE IF NOT EXISTS test_table (
171         id SERIAL PRIMARY KEY,
172         name VARCHAR(255) NOT NULL,
173         age INTEGER
174     );
175 """)
176
177 # Commit the changes
178 conn.commit()
179
180 # Insert data into the table
181 cur.execute("""
182     INSERT INTO test_table (name, age)
183     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
184 """)
185
186 # Commit the changes
187 conn.commit()
188
189 # Query the data
190 cur.execute("""
191     SELECT * FROM test_table;
192 """)
193
194 # Fetch all rows
195 rows = cur.fetchall()
196
197 # Print the results
198 for row in rows:
199     print(row)
200
201 # Close the cursor and connection
202 cur.close()
203 conn.close()
204
205 # Test client(s)
206
207 # Create a connection to the PostgreSQL server
208 conn = psycopg2.connect(
209     host="localhost",
210     port=5432,
211     user="postgres",
212     password="postgres",
213     database="postgres"
214 )
215
216 # Create a cursor object
217 cur = conn.cursor()
218
219 # Create a table
220 cur.execute("""
221     CREATE TABLE IF NOT EXISTS test_table (
222         id SERIAL PRIMARY KEY,
223         name VARCHAR(255) NOT NULL,
224         age INTEGER
225     );
226 """)
227
228 # Commit the changes
229 conn.commit()
230
231 # Insert data into the table
232 cur.execute("""
233     INSERT INTO test_table (name, age)
234     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
235 """)
236
237 # Commit the changes
238 conn.commit()
239
240 # Query the data
241 cur.execute("""
242     SELECT * FROM test_table;
243 """)
244
245 # Fetch all rows
246 rows = cur.fetchall()
247
248 # Print the results
249 for row in rows:
250     print(row)
251
252 # Close the cursor and connection
253 cur.close()
254 conn.close()
255
256 # Test client(s)
257
258 # Create a connection to the PostgreSQL server
259 conn = psycopg2.connect(
260     host="localhost",
261     port=5432,
262     user="postgres",
263     password="postgres",
264     database="postgres"
265 )
266
267 # Create a cursor object
268 cur = conn.cursor()
269
270 # Create a table
271 cur.execute("""
272     CREATE TABLE IF NOT EXISTS test_table (
273         id SERIAL PRIMARY KEY,
274         name VARCHAR(255) NOT NULL,
275         age INTEGER
276     );
277 """)
278
279 # Commit the changes
280 conn.commit()
281
282 # Insert data into the table
283 cur.execute("""
284     INSERT INTO test_table (name, age)
285     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
286 """)
287
288 # Commit the changes
289 conn.commit()
290
291 # Query the data
292 cur.execute("""
293     SELECT * FROM test_table;
294 """)
295
296 # Fetch all rows
297 rows = cur.fetchall()
298
299 # Print the results
300 for row in rows:
301     print(row)
302
303 # Close the cursor and connection
304 cur.close()
305 conn.close()
306
307 # Test client(s)
308
309 # Create a connection to the PostgreSQL server
310 conn = psycopg2.connect(
311     host="localhost",
312     port=5432,
313     user="postgres",
314     password="postgres",
315     database="postgres"
316 )
317
318 # Create a cursor object
319 cur = conn.cursor()
320
321 # Create a table
322 cur.execute("""
323     CREATE TABLE IF NOT EXISTS test_table (
324         id SERIAL PRIMARY KEY,
325         name VARCHAR(255) NOT NULL,
326         age INTEGER
327     );
328 """)
329
330 # Commit the changes
331 conn.commit()
332
333 # Insert data into the table
334 cur.execute("""
335     INSERT INTO test_table (name, age)
336     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
337 """)
338
339 # Commit the changes
340 conn.commit()
341
342 # Query the data
343 cur.execute("""
344     SELECT * FROM test_table;
345 """)
346
347 # Fetch all rows
348 rows = cur.fetchall()
349
350 # Print the results
351 for row in rows:
352     print(row)
353
354 # Close the cursor and connection
355 cur.close()
356 conn.close()
357
358 # Test client(s)
359
360 # Create a connection to the PostgreSQL server
361 conn = psycopg2.connect(
362     host="localhost",
363     port=5432,
364     user="postgres",
365     password="postgres",
366     database="postgres"
367 )
368
369 # Create a cursor object
370 cur = conn.cursor()
371
372 # Create a table
373 cur.execute("""
374     CREATE TABLE IF NOT EXISTS test_table (
375         id SERIAL PRIMARY KEY,
376         name VARCHAR(255) NOT NULL,
377         age INTEGER
378     );
379 """)
380
381 # Commit the changes
382 conn.commit()
383
384 # Insert data into the table
385 cur.execute("""
386     INSERT INTO test_table (name, age)
387     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
388 """)
389
390 # Commit the changes
391 conn.commit()
392
393 # Query the data
394 cur.execute("""
395     SELECT * FROM test_table;
396 """)
397
398 # Fetch all rows
399 rows = cur.fetchall()
400
401 # Print the results
402 for row in rows:
403     print(row)
404
405 # Close the cursor and connection
406 cur.close()
407 conn.close()
408
409 # Test client(s)
410
411 # Create a connection to the PostgreSQL server
412 conn = psycopg2.connect(
413     host="localhost",
414     port=5432,
415     user="postgres",
416     password="postgres",
417     database="postgres"
418 )
419
420 # Create a cursor object
421 cur = conn.cursor()
422
423 # Create a table
424 cur.execute("""
425     CREATE TABLE IF NOT EXISTS test_table (
426         id SERIAL PRIMARY KEY,
427         name VARCHAR(255) NOT NULL,
428         age INTEGER
429     );
430 """)
431
432 # Commit the changes
433 conn.commit()
434
435 # Insert data into the table
436 cur.execute("""
437     INSERT INTO test_table (name, age)
438     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
439 """)
440
441 # Commit the changes
442 conn.commit()
443
444 # Query the data
445 cur.execute("""
446     SELECT * FROM test_table;
447 """)
448
449 # Fetch all rows
450 rows = cur.fetchall()
451
452 # Print the results
453 for row in rows:
454     print(row)
455
456 # Close the cursor and connection
457 cur.close()
458 conn.close()
459
460 # Test client(s)
461
462 # Create a connection to the PostgreSQL server
463 conn = psycopg2.connect(
464     host="localhost",
465     port=5432,
466     user="postgres",
467     password="postgres",
468     database="postgres"
469 )
470
471 # Create a cursor object
472 cur = conn.cursor()
473
474 # Create a table
475 cur.execute("""
476     CREATE TABLE IF NOT EXISTS test_table (
477         id SERIAL PRIMARY KEY,
478         name VARCHAR(255) NOT NULL,
479         age INTEGER
480     );
481 """)
482
483 # Commit the changes
484 conn.commit()
485
486 # Insert data into the table
487 cur.execute("""
488     INSERT INTO test_table (name, age)
489     VALUES ('John', 30), ('Jane', 25), ('Bob', 35);
490 """)
491
492 # Commit the changes
493 conn.commit()
494
495 # Query the data
496 cur.execute("""
497     SELECT * FROM test_table;
498 """)
499
500 # Fetch all rows
501 rows = cur.fetchall()
502
503 # Print the results
504 for row in rows:
505     print(row)
506
507 # Close the cursor and connection
508 cur.close()
509 conn.close()
510
511 # Test client(s)
512
513 # Create a connection to the PostgreSQL server
514 conn = psycopg2.connect(
515     host="localhost",
516     port=5432,
517     user="postgres",
518     password="postgres",
519     database="postgres"
520 )
521
522 # Create a
```

Fitur:

Menambahkan tampilan halaman web admin-dashboard untuk menambah data laptop.



Fitur:

Menambahkan tampilan halaman web admin-dashboard untuk mengubah data laptop.