

Your Github URL

Your Vercel URL

W14-P1: show first welcome page

=>

```

index.ejs
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title><%= title %></title>
7   </head>
8   <body>
9     <h1><%= title %></h1>
10    <h2>Welcome to <%= title %></h2>
11    <h3><%= name %>, <%= id %></h3>
12    <footer>
13      <p>&copy; 2025 TKU</p>
14    </footer>
15  </body>
16</html>

server_14.js
1 import express from "express";
2
3 const app_14 = express();
4
5 app_14.use(express.static("public"));
6 app_14.set("view engine", "ejs");
7
8 app_14.use("/", (req, res, next) => {
9   res.render("index", {
10     title: "Express Server",
11     name: "Liang-Ting Lin",
12     id: "913410014"
13   });
14 });
15
16 const port = process.env.PORT || 5000;
17
18 app_14.listen(port, () => {
19   console.log(`Server is running on port ${port}`);
20 });

```

4554eaf%09zero2005x%09Thu May 22 19:50:26 2025 +0800 W14-P1: show first welcome page

W14-P2: Show static products page

=> show render page with data passed into ejs page

```

1 import express from "express";
2
3 const app_14 = express();
4
5 app_14.use(express.static("public"));
6 app_14.set("view engine", "ejs");
7
8 app_14.use("/product_14/static", (req, res, next) => {
9   console.log("static");
10  res.render('product_14/static', {
11    title: "Get Product - Static",
12    name: "Liang-Ting Lin",
13    id: "913410014",
14  });
15});
16
17 app_14.use("/", (req, res, next) => {
18  res.render("index", {
19    title: "Express Server",
20    name: "Liang-Ting Lin",
21    id: "913410014",
22  });
23});
24
25 const port = process.env.PORT || 5000;
26
27 app_14.listen(port, () => {
28   console.log(`Server is running on port ${port}`);
29 });
30

```

Get Product - Static  
Liang-Ting Lin, 913410014

1-Emperor Bed  
\$21.99

=> show how the product-1.jpg can be accessed fro public directory as root directory

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <link rel="stylesheet" href="/style.css" />
7   </head>
8   <body>
9     <div style="text-align: center; margin-top: 2rem">
10       <h2>%= title %</h2>
11       <h4 style="margin-top: -1.5rem">%= name %, <%= id %</h4>
12     </div>
13     <div class="products-container">
14       <div class="single-product">
15         
21       </div>
22       <div class="single-product">
23         
29     </div>
30   </div>
31   <div style="text-align: center; margin-top: 2rem">
32     <h3>Buy Now</h3>
33     <form>
34       <input type="button" value="Buy Now" />
35     </form>
36   </div>
37 </body>
38 </html>

```

Get Product - Static  
Liang-Ting Lin, 913410014

1-Emperor Bed  
\$21.99

c19ed13%09zero2005x%09Thu May 22 19:54:09 2025 +0800 W14-P2: Show static products page

W14-P3: Create database js\_demo\_xx, table product\_xx, and insert 12 data, write test code to get all products

=> Create database js\_demo\_xx, table product\_xx, and insert 12 data

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer sidebar displays the database structure, including servers, databases (js\_demo\_14), and tables (product\_14). The main window shows a query editor with the SQL command `select * from public.product_14` and a data output grid displaying 12 rows of product data. A status bar at the bottom indicates "Table rows counted: 12".

	<code>id [PK] integer</code>	<code>title character varying (255)</code>	<code>price real</code>	<code>category character varying (255)</code>	<code>img text</code>	<code>remote_img text</code>
1	1	Emperor Bed	21.99	Liddy	./images/product-1.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
2	2	Accent Chair	25.99	Caressa	./images/product-2.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
3	3	High-Back Bench	9.99	Ikea	./images/product-3.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
4	4	Wooden Table	19.99	Ikea	./images/product-4.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
5	5	Dining Table	69.99	Caressa	./images/product-5.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
6	6	Entertainmint Center	25.99	Liddy	./images/product-6.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
7	7	Albany Sectional	10.99	Ikea	./images/product-7.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
8	8	Sofa Set	69.99	Liddy	./images/product-8.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
9	9	Utopia Sofa	69.99	Liddy	./images/product-9.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
10	10	Modern Bookshelf	8.99	Marcos	./images/product-10.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
11	11	Albany Table	79.99	Marcos	./images/product-11.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p
12	12	Leather Sofa	39.95	Liddy	./images/product-12.jpg	https://loxlxwyirrqkjrruqbq.supabase.co/storage/v1/object/public/htchung-1n-14/p

=> connect js\_demo\_14 database

The screenshot shows a VS Code terminal session connected to a Node.js environment. The user has run the command `node ./database.js`, which logs the connection details and the product data from the database. The terminal output is as follows:

```

idleTimeoutMillis: 10000
}
results [{"current_time":"2025-05-22T12:42:35.309Z"}]
(base) PS C:\Users\liangtinglin\Documents\淡江課程\javascript課程\113-2-IN-DEMO-14\demo\w14_fullstack_14\server_14\utilis> node ./database.js
Connecting to the database...
host: 'localhost',
port: 5432,
min: 0,

```

At the same time, the pgAdmin 4 interface shows the same product data grid, confirming the connection.

=> get all product data

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure for "1132-IN-DEMO-14" containing files like demo, w04\_basics\_14, w05\_basics\_14, w06\_prep\_test\_14, w10\_mid1\_m3\_14, w10\_product\_superbase..., w11\_async\_js\_14, w12\_ajax\_async\_14, w13\_fetch\_async\_14, w13\_product\_api\_14, w14\_fullstack\_14, client\_14, server\_14, node\_modules, public, routers/api, JS apiProductRouter..., utils, data, database.js, and db-testjs.
- Terminal:** Displays a command-line session in PowerShell (Windows) showing the execution of a Node.js script named db-test.js. The script imports db from database.js and attempts to connect to a database using the db.query("SELECT \* FROM product\_14") method. It logs the results to the console and handles errors.
- Output:** Shows the results of the database query, listing various products with their IDs, titles, prices, categories, and remote image URLs.
- Bottom Status Bar:** Includes status indicators for Ln 5, Col 64, Spaces: 4, UTF-8, CRLF, and file type JavaScript.

```

1 import db from "./database.js";
2
3 const testDB = async () => {
4   try {
5     const results = await db.query("SELECT * FROM product_14");
6     console.log("results", JSON.stringify(results.rows));
7   } catch (error) {
8     console.error("Error testing the database connection", error);
9   }
10 };
11
(base) PS C:\Users\liantinglin\Documents\淡江課程\javascript課程\1132-IN-DEMO-14\demo>w14_fullstack_14\server_14\utils> node .\db-test.js
results [{"id":1,"title":"Emperor Bed","price":21.99,"category":"Liddy","img":"/images/product-1.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-1.jpg"}, {"id":2,"title":"Accent Chair","price":25.99,"category":"Caressa","img":"/images/product-2.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-2.jpg"}, {"id":3,"title":"High Back Bench","price":9.99,"category":"Ikea","img":"/images/product-3.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-3.jpg"}, {"id":4,"title":"Wooden Table","price":19.99,"category":"Ikea","img":"/images/product-4.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-4.jpg"}, {"id":5,"title":"Dining Table","price":69.99,"category":"Caressa","img":"/images/product-5.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-5.jpg"}, {"id":6,"title":"Entertainment Center","price":25.99,"category":"Liddy","img":"/images/product-6.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-6.jpg"}, {"id":7,"title":"Albany Sectional","price":10.99,"category":"Ikea","img":"/images/product-7.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-7.jpg"}, {"id":8,"title":"Sofa Set","price":69.99,"category":"Liddy","img":"/images/product-8.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-8.jpg"}, {"id":9,"title":"Utopia Sofa","price":69.99,"category":"Liddy","img":"/images/product-9.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-9.jpg"}, {"id":10,"title":"Modern Bookshelf","price":8.99,"category":"Marcos","img":"/images/product-10.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-10.jpg"}, {"id":11,"title":"Albany Table","price":79.99,"category":"Marcos","img":"/images/product-11.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-11.jpg"}, {"id":12,"title":"Leather Sofa","price":39.95,"category":"Liddy","img":"/images/product-12.jpg","remote_img":"https://loxlxwirrqkjrrqboq.supabase.co/storage/v1/object/public/hutchung-in-14/product_14/product-12.jpg"}]

```