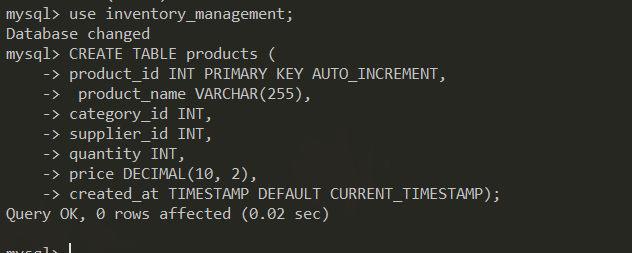
**Task 1: Database and Tables Creation**

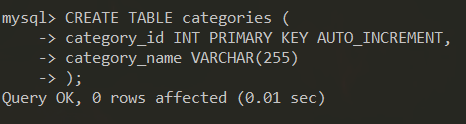
Q1: Create a database named inventory\_management.



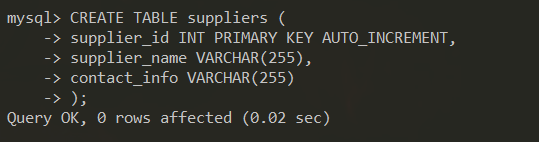
Q2: Create a products table with the following columns: product\_id, product\_name, category\_id, supplier\_id, quantity, price, and created\_at.



Q3: Create a categories table with the following columns: category\_id and category\_name.

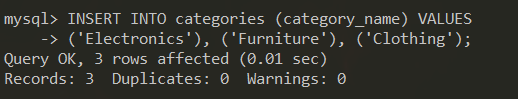


Q4: Create a suppliers table with the following columns: supplier\_id, supplier\_name, and contact\_info.

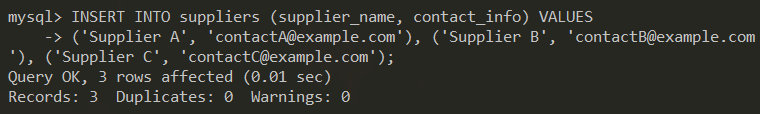


**Task 2: Data Insertion and Updating**

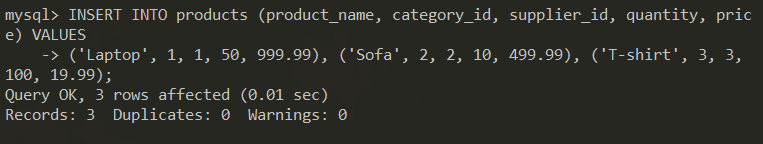
**Q1: Insert sample data into the** categories **table.**



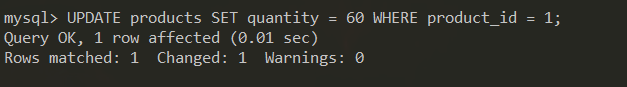
Q2: Insert sample data into the suppliers table.



Q3: Insert sample data into the products table.

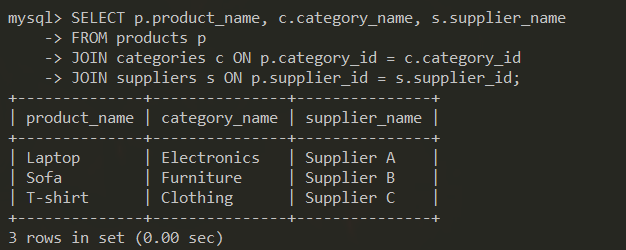


Q4: Update the quantity of the product with product\_id 1 to 60



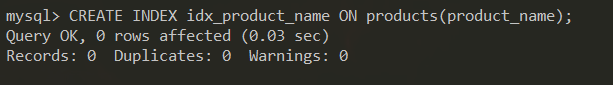
**Task 3: Joins**

Q1: Write a query to retrieve all products along with their category names and supplier names.



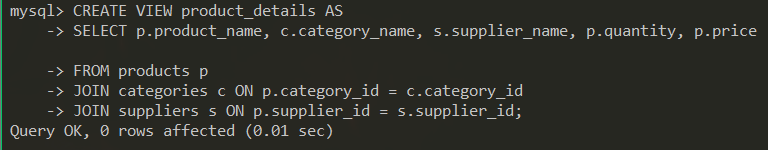
**Task 4: Indexes**

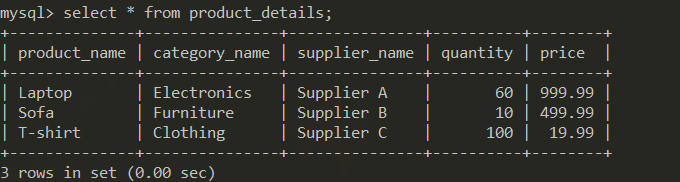
Q1: Create an index on the product\_name column of the products table.



**Task 5: Views**

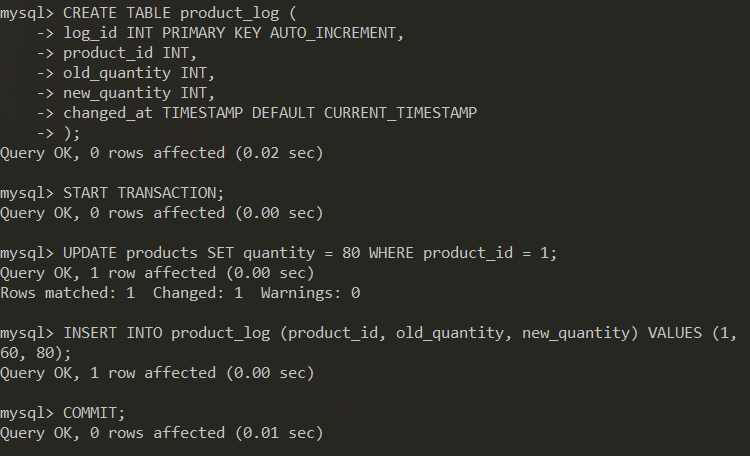
Q1: Create a view named product\_details that displays product\_name, category\_name, supplier\_name, quantity, and price.





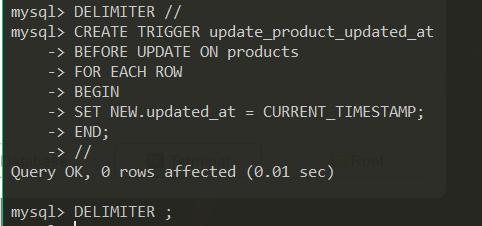
**Task 6: Transactions**

Q1: Write a transaction to update the quantity of a product and log the change in a product\_log table. Create the product\_log table first.



**Task 7: Triggers**

Q1: Create a trigger to automatically update the updated\_at column of the products table whenever a row is updated.



**Frontend Tasks**

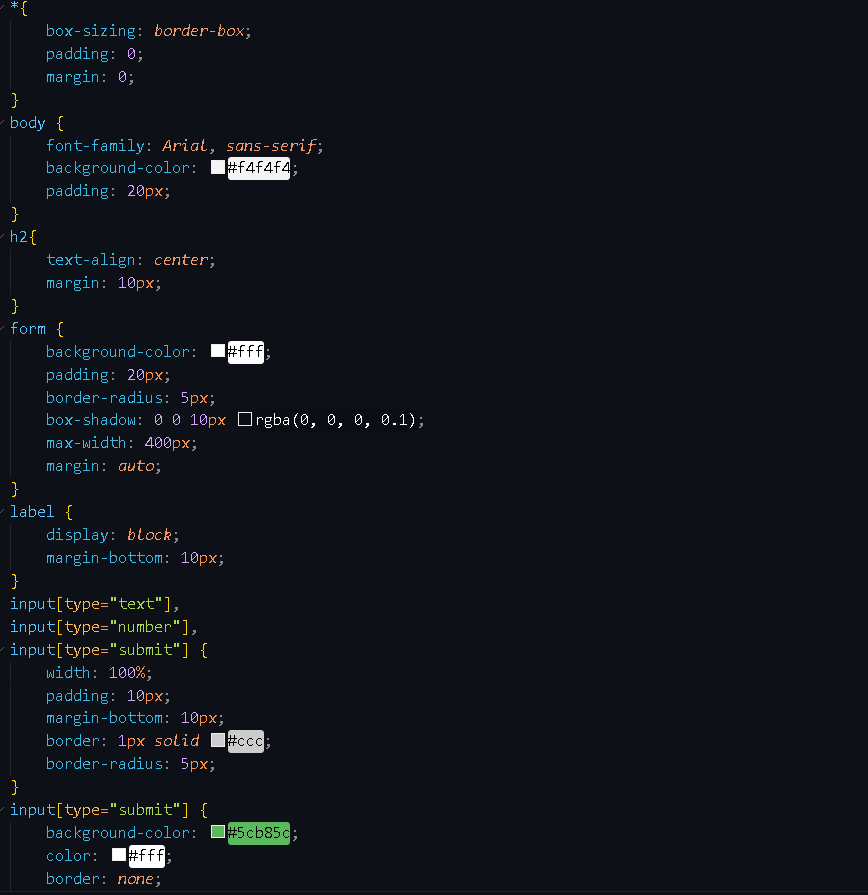
**Task 8: HTML**

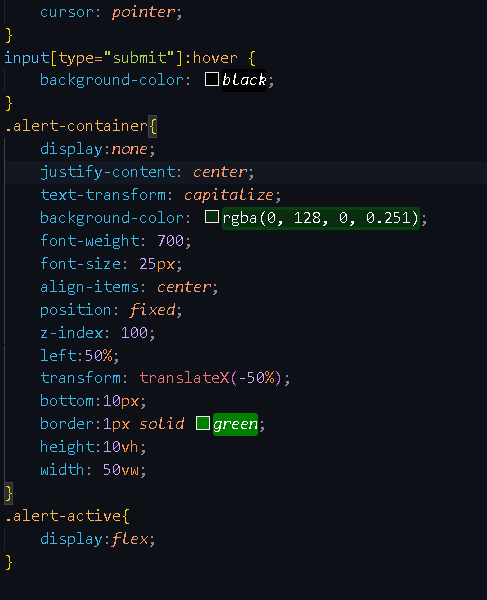
Q1: Create an HTML file named index.html that includes a form to add a new product.



**Task 9: CSS**

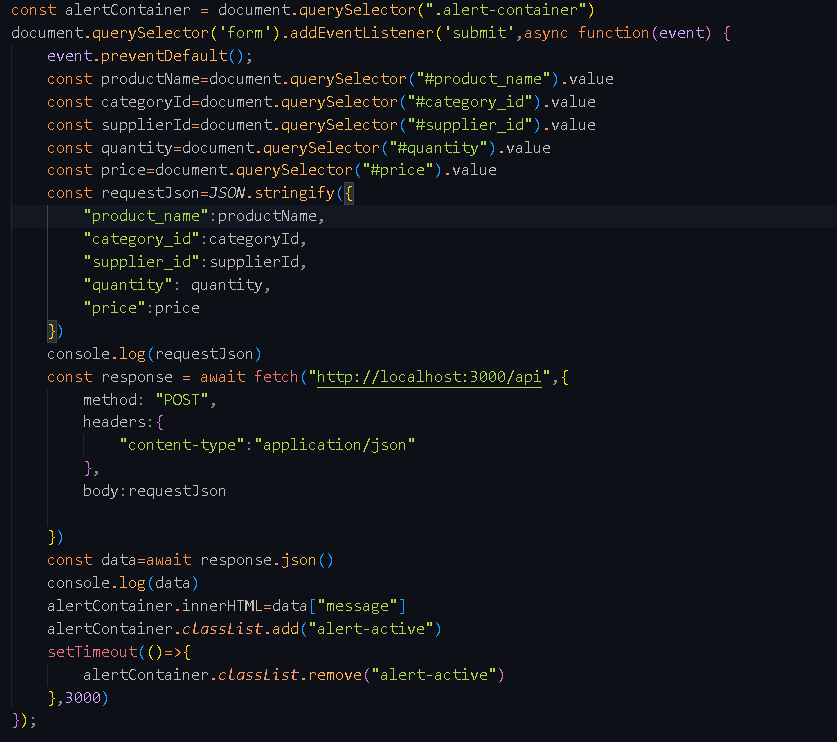
Q1: Create a CSS file named styles.css to style the form created in the HTML file.





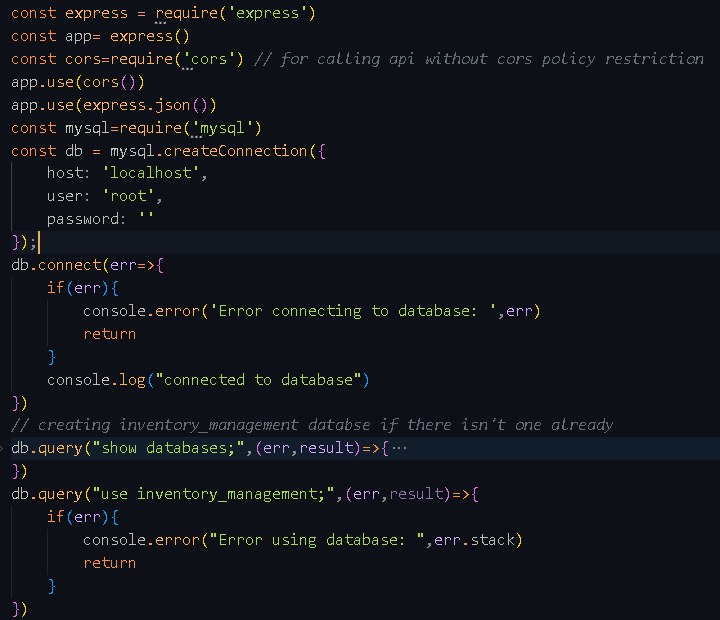
**Task 10: JavaScript**

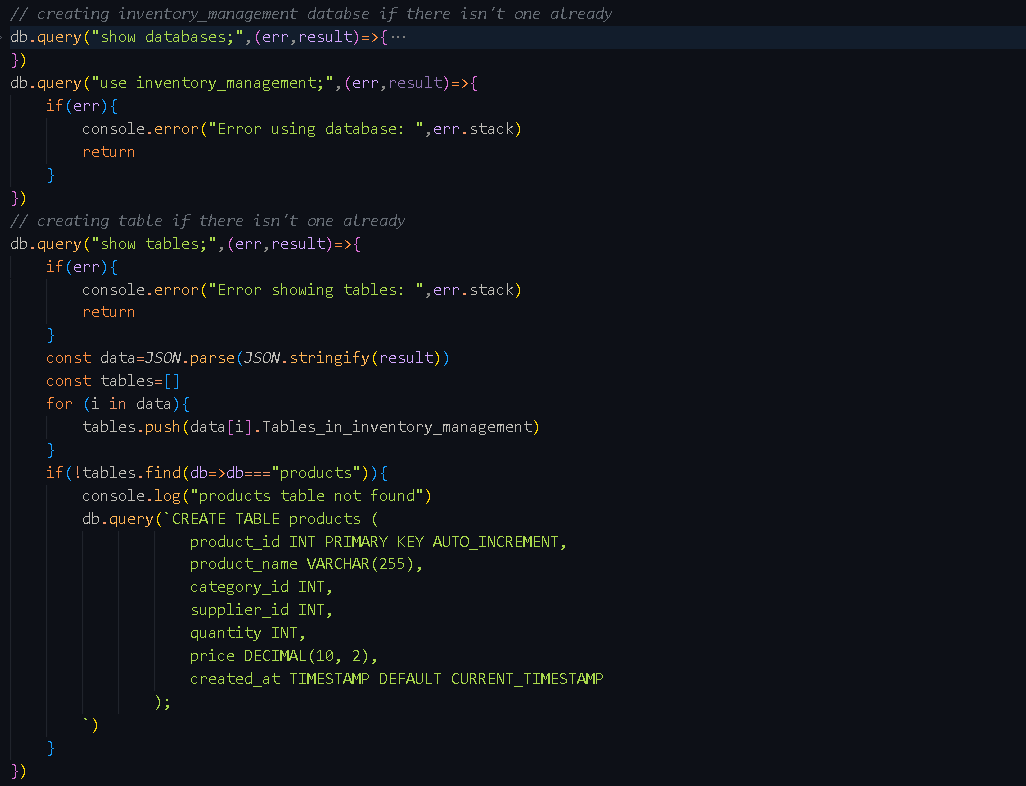
Q1: Create a JavaScript file named script.js to handle the form submission and send the data to the server using AJAX.

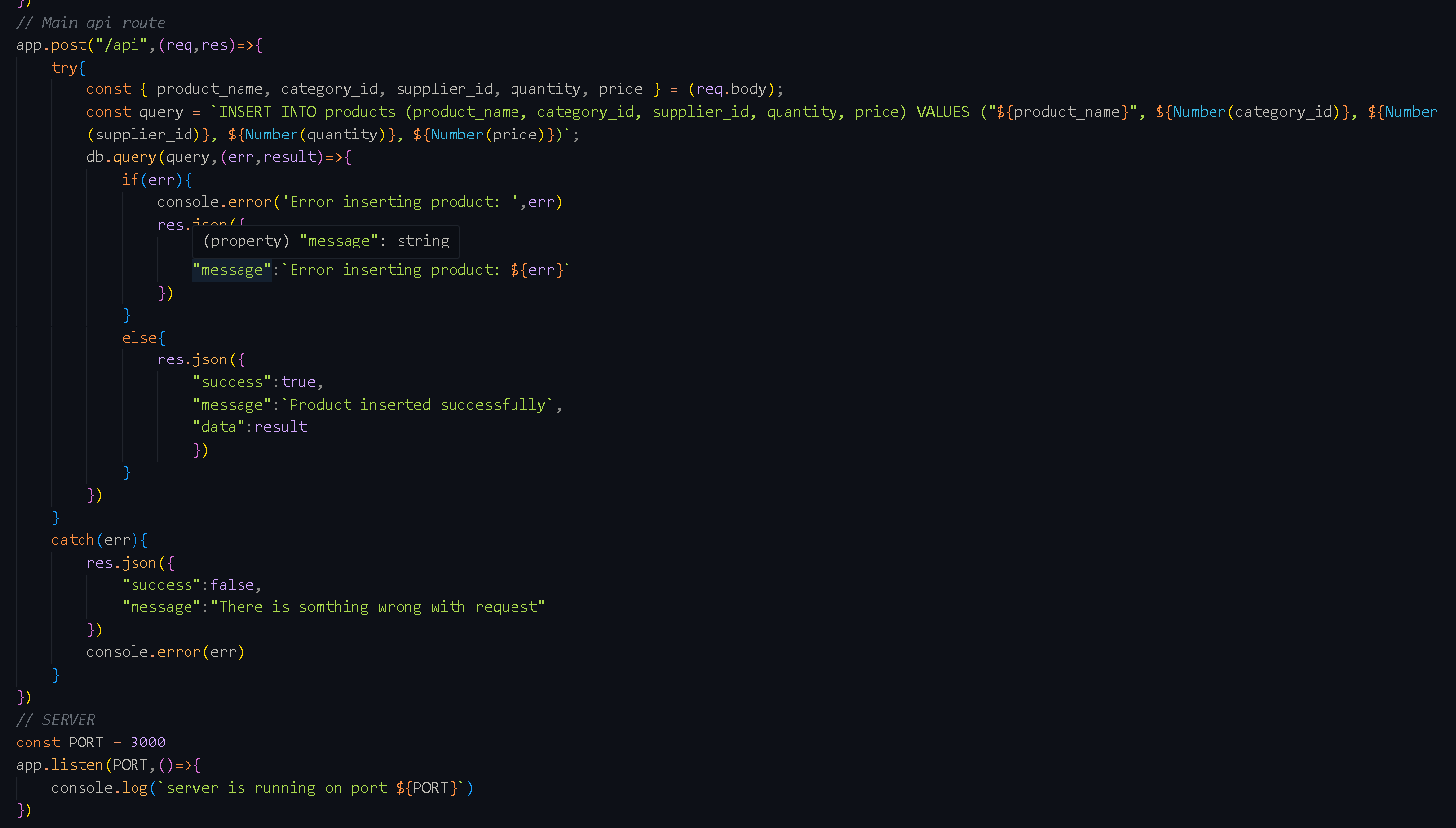


**Task 11: Backend (Integration)**

Q1: Write a backend script in your preferred language (e.g., PHP, Node.js) to handle the product addition request from the form.



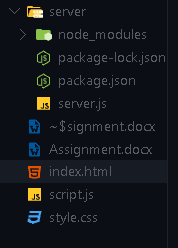




**Final Integration**

**Q1: Integrate the HTML, CSS, and JavaScript files into a single project directory.**

* Place index.html, styles.css, and script.js in the root directory.



* Ensure the form in index.html uses the correct paths to link the CSS and JS files:



