LWC Component: Fetch Price by Product Name

1. Apex Controller

```
public with sharing class ElectronicProductController {
    @AuraEnabled(cacheable=true)
    public static ElectronicProduct__c getProductByName(String name) {
        List<ElectronicProduct__c> products = [
            SELECT Id, Name, Price__c
            FROM ElectronicProduct__c
            WHERE Name = :name
            LIMIT 1
        ];
        if (products.isEmpty()) {
            return null;
        }
        return products[0];
    }
}
```

2. productPriceFetcher.html

```
<template>
    <lightning-card title="Product Price Fetcher" icon-name="custom:custom63">
        <div class="slds-p-around_medium">
            <lightning-input
                label="Enter Product Name"
                value={productName}
                onchange={handleNameChange}>
            </lightning-input>

            <lightning-button
                label="Get Price"
                variant="brand"
                class="slds-m-top_small"
                onclick={fetchPrice}>
            </lightning-button>

            <template if:true={product}>
                <div class="slds-m-top_medium">
                    <p><strong>Name:</strong> {product.Name}</p>
                    <p><strong>Price:</strong> INR {product.Price__c}</p>
                </div>
            </template>

            <template if:true={errorMessage}>
                <p class="slds-text-color_error slds-m-top_medium">{errorMessage}</p>
            </template>
        </div>
    </lightning-card>
</template>
```

LWC Component: Fetch Price by Product Name

3. productPriceFetcher.js

```
import { LightningElement, track } from 'lwc';
import getProductByName from '@salesforce/apex/ElectronicProductController.getProductByName';

export default class ProductPriceFetcher extends LightningElement {
    @track productName = '';
    @track product;
    @track errorMessage;

    handleNameChange(event) {
        this.productName = event.target.value;
    }

    fetchPrice() {
        if (!this.productName) {
            this.errorMessage = 'Please enter a product name.';
            this.product = null;
            return;
        }

        getProductByName({ name: this.productName })
            .then(result => {
                if (result) {
                    this.product = result;
                    this.errorMessage = null;
                } else {
                    this.product = null;
                    this.errorMessage = 'Product not found.';
                }
            })
            .catch(error => {
                this.product = null;
                this.errorMessage = 'Error: ' + error.body.message;
            });
    }
}
```

4. productPriceFetcher.js-meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>59.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
        <target>lightning__AppPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```