1. **Created Necessary Objects and Fields in Salesforce**

**Invoice Object**

* **Object Name**: Invoice
* **Fields**:
  + **Account\_\_c** (Lookup to Account)
  + **Invoice\_Number\_\_c** (Text)
  + **Invoice\_Date\_\_c** (Date)
  + **Due\_Date\_\_c** (Date)
  + **Invoice\_Reference\_\_c** (invoice)
  + **Total\_\_c** (Currency)

**Invoice Line Item Object**

* **Object Name**: Invoice Line Item
* **Fields**:
  + **Invoice\_\_c** (Lookup to Invoice)
  + **Line\_Description\_\_c** (Text)
  + **Quantity\_\_c** (Number)
  + **Unit\_Price\_\_c** (Currency)
  + **Amount\_\_c** (Formula: Quantity\_\_c \* Unit\_Price\_\_c)

1. **Created Apex Controller**

**Apex Controller (CreateInvoiceController)**

The Apex class will handle URL parameters, dynamically query child records, process field mappings, and create the **Invoice** and related **Invoice Line Items**.

public with sharing class InvoiceController

@AuraEnabled

public static Id createInvoiceAndLineItems(Map<String, string> inputData) {

try {

String accountId = (String)inputData.get('accountId');

String invoiceDateStr = (String)inputData.get('invoiceDate');

String dueDateStr = (String)inputData.get('dueDate');

Date invoiceDate = parseDate(invoiceDateStr);

Date dueDate = parseDate(dueDateStr);

Invoice\_\_c invoice = new Invoice\_\_c();

invoice.Account\_\_c = accountId;

invoice.Invoice\_Date\_\_c = invoiceDate;

invoice.Due\_Date\_\_c = dueDate;

insert invoice;

return invoice.Id;

} catch (Exception e) {

throw new AuraHandledException('Error creating Invoice and Line Items: ' + e.getMessage());

}

}

private static Date parseDate(String dateStr) {

if (String.isNotBlank(dateStr)) {

try {

if (dateStr.contains('-')) {

return Date.valueOf(dateStr);

}

List<String> dateParts = dateStr.split('/');

if (dateParts.size() == 3) {

String reformattedDateStr = dateParts[2] + '-' + dateParts[1] + '-' + dateParts[0];

return Date.valueOf(reformattedDateStr);

}

} catch (Exception e) {

throw new AuraHandledException('Invalid date format: ' + dateStr);

}

}

throw new AuraHandledException('Date cannot be blank');

}

}  
  
  
  
**3. Created a Lightning Web Component (LWC) to Display URL Parameters**

In the Lightning Web Component, display the URL parameters in a table for verification and provide a button to create the invoice.  
  
  
**4. Created Custom Button on Opportunity/Order**

* Go to **Opportunity** or **Order** object in Salesforce Setup.
* Under **Buttons, Links, and Actions**, create a new **Detail Page Button**.
  + **Behavior**: Display in a new window or redirect.
  + **URL**: /lightning/n/create\_invoice1  
    ?c\_\_origin\_record={!Opportunity.Id}  
    &c\_\_account={!Opportunity.AccountId}  
    &c\_\_invoice\_date={!Opportunity.CloseDate}  
    &c\_\_invoice\_due\_date={!Opportunity.invoice\_due\_date\_\_c}  
    &c\_\_child\_relationship\_name= {!Opportunity.ContractId}  
    &c\_\_line\_item\_description= {!Opportunity.Description}  
    &c\_\_line\_item\_quantity={!Opportunity.TotalOpportunityQuantity}  
    &c\_\_line\_item\_unit\_price={!Opportunity.unit\_price\_\_c}

