1. Project details
   - Name of project: Smart Invoice
   - Team ID: 32
   - Team member ID: YIWEN CAI(30648449), YINGTAO LIU(29988259), YUJIE WANG(30017429)
   - Team Member Username: ycai21/yliu157/ywang240

2. Problem Statement
   - Problem-solving: We propose an invoice management system that helps sales to create/check/update invoice information.
   - Why not Excel: invoice created will eventually be stored together. Using an Excel will create unnecessary problems when one tries to create a new invoice. For example, one may create a duplicate invoice or mistakenly update the wrong invoice.
     The large amount of invoice and multiple update in real time also create difficulties if used spreadsheet storage form.

3. Target user
   - Who will use the database: Salesman
   - Who will administer the database: Sales Manager

In reality, sales will use the database to search for specific information about customers, orders, and invoices. The sales manager is responsible for managing the database, such as updating new invoices and customers.

4. List of Relations:

Customer			
SSN (VARCHAR 15)	Cust_Name(VARCHAR 30)	Cust_Address (VARCHAR 255)	Pho_number (Varchar 15)
Order			
Order_Num (INT 20)	Invoice_ID(VARCHAR 20)	Product_ID (VARCHAR 25)	Product_Quantity (INT 20)

Product		
Pro_ID (VARCHAR 25)	Pro_Name (VARCHAR 30)	Pro_Price (DECIMAL (10,2))

Invoice			
Inv_ID (VARCHAR 20)	Inv_Date (DATE())	Order_Number (INT 20)	Cust_SSN (VARCHAR 15)

*The attribute with underline is the primary key for that relation.

Cust_Name represents customer name; Cust_Address represents customer address; Pho_number represent customer's phone number.
Order_Num represents order number;
Inv_ID represents invoice ID; Inv_Date represents the date of the invoice; Cust_SSN represents customers' SSN
Pro_ID represents ID of the product; Pro_name is the name of the product; Pro_Price is the price of the product in dollars.
SSN is the primary key of Customer. Order_Num is the primary key of Order Inv_ID is the primary key of Invoice. Pro_ID is the primary key of the product.

5. Web-interface:
https://projects.invisionapp.com/prototype/DB-Projoect-Web-Interface-Demo-ck0svv3vc0015we01qas9c3oa/play/bc8f7d4e

Once click on the link, you would be prompted to what is a prototype of the web interface. What you will see at the beginning is the idea of main page. One can choose to add a new invoice or search invoice by the information on hand. Once click on the add new bottom, the user will be asked to input all the information on the invoice. Once click on add, the system will ideally check the input to see if any database integrity law is violated. (eg, duplicate invoice id, etc)If not, the invoice will be successfully added to the database and the user will receive a prompt and go back to the main page. Once enter the information and click the search button, a list that contains all the invoices that match with the information given will be shown. The list will contain some brief information regarding the invoice to help users identify whether or not it is what they are looking for with efficiency. The user can also choose to view details of the invoice, adding the invoice information, and delete the invoice by clicking on the corresponding bottom.
Notice that the demo is designed only to demonstrate the most foundational idea of the web interface, and only partial functions are implemented in the demo.

6. Data:

We will use arbitrary data that generated from some sort of random function for the testing purpose at the beginning. Then we will try to gather some real world data from some merchants if possible. Then we will modify the data a little to make the data compatible with the relation we create.