## INVOICE APPLICATION

## DOCUMENTATION

## GLOBAL -AI TECHNOLGIES

## **~** By Nomula Sri Durga

## [sridurga3nomula@gmail.com](mailto:sridurga3nomula@gmail.com)

## +91 7382479664

## Geethanjali college of engineering and Technologies

## ABSTRACT

This project brings forth an Invoice Application created to ease and streamline the process of preparing professional invoices for business deals. The system presents a user-friendly interface in which users can dynamically include or exclude several items, enter quantities, rates, discounts, and corresponding GST percentages, and get immediate totals and grand total calculations. For enhanced usability, the application features smart auto complete options for inputs like buyer information, vehicle numbers, and invoice numbers, with easy access to commonly required data. It also provides more accuracy by typing out numerical values in words based on the Indian system of numbering, which is predominantly adopted in financial papers. Auto-numbering of invoices and persistent data storage facilitate orderly keeping of records and ease the retrieval of previous invoices. Additionally, the program comes with a print-ready design, which means that the invoices created can be easily printed in a professional format without needing to be reformatted. As a user-friendly, dependable, and effective tool, this system is a convenient utility for small- to medium-sized enterprises to effectively carry out their invoicing requirements.

## TABLE OF CONTENTS

[Abstract](#_bookmark0) I

[Table of Contents](#_TOC_250034) II

1. [INTRODUCTION 1](#_TOC_250033)
2. OBJECTIVES 2
3. PROPOSED SYSTEM 4
4. SOFTWARE REQUIREMENTS 7
5. IMPLEMENTATION 8
6. OUTPUT SCREENS 27
7. CONCLUSION 31

## INTRODUCTION

In the current fast-paced business world, accurate and efficient billing is a key necessity for businesses of all sizes. The act of creating invoices not only creates an official record of a financial transaction but also helps in ensuring transparency, accountability, as well as trust between clients and businesses. Manually, invoice preparation has traditionally been a time-consuming task, tending to introduce errors in calculations, duplications of records, and challenges in keeping a systematic archive of transactions. To address these weaknesses, companies are turning more towards automated invoicing systems that can streamline operations but enhance reliability and accuracy.

This project, Invoice Application, has been created to solve these issues by offering an easy and yet effective platform for creating professional invoices. The system provides a user-friendly interface where users can simply enter information such as buyer details, product or service description, quantity, rate, applicable discount, and GST percentage. When the data is entered, the application automatically calculates totals, calculates super discounts, and comes up with the final amount payable with high accuracy. To make it even more transparent, the total amount is also written in words in the Indian numbering system, making the invoice acceptable for business purposes legally and professionally.

The application also includes various features that enhance efficiency and user convenience. Intelligent autocomplete feature enables users to easily complete recurring information like customer names, addresses, and vehicle numbers from past records. Automated invoice numbering ensures proper systematic maintenance of records, while persistent storage enables users to access previous invoices whenever they need to. Furthermore, the system has been configured in a print-ready format, allowing users to produce neat, professional-looking invoices that can be handed over to clients directly without further formatting.

**With accuracy, automation, and simplicity, this Invoice Application streamlines the billing process for small and medium-scale businesses. It reduces errors, saves time, and ensures invoices are professional and compliant with business standards.**

## OBJECTIVES

1. **To develop and integrate a simple invoicing system**

The main aim of the project is to develop a system to make the process of making invoices simpler. The interface was made simple so that users can input business information, buyer details, and product or service descriptions easily without needing technical skills.

1. **To make automatic calculations for totals, discounts, GST, and net amounts**

Manual computation in billing tends to cause inaccuracies and inefficiencies. This project makes all the sums, discounts, GST rates, and the final amount payable automatically computed by the system. This reduces human error and assures accurate billing.

1. **To give amount conversion in words**

Invoices in India need the final amount to be visible in both numbers and words for legal and professional reasons. The system automatically translates the numerical value into words based on the Indian numbering system (Thousand, Lakh, Crore) to ensure compliance and clarity.

1. **For implementing autocomplete functionality for quicker input**

Entering buyer information or reused data is time-consuming. The software has smart autocomplete, which retains values previously entered for such as buyer names, addresses, and vehicle numbers. This not only makes data entry fast but also ensures consistency on invoices.

1. **For systematic invoice numbering**

Having a proper order of invoice numbers is important for accounting and auditing. The project features an automatic invoice numbering that automatically increments the number sequentially after each generated invoice. This prevents duplication and proper record keeping.

1. **To provide persistent storage and retrieval of invoices**

Another significant goal is enabling users to save invoices for later use. Through storage facilities, the program makes it possible for previously created invoices to be saved, accessed, and reused whenever needed. This enhances convenience and facilitates keeping historical records.

1. **To provide a print-ready invoice format**

A professional invoice must be neat, well-organized, and print-ready. The system is implemented with a print-friendly design such that invoices could be created and printed out without the need for further formatting. This eliminates time and guarantees documents appear consistent.

1. **To minimize human errors, save time, and streamline the billing process**

Generally, the project seeks to minimize dependence on manual labor in billing. Through automation of critical processes like calculations, storing records, and formatting invoices, the system eliminates human mistakes, enhances precision, and saves a lot of time. This makes billing more efficient, particularly for small and medium-sized enterprises.

## PROPOSED SYSTEM

* **Features Of Proposed system:**

1. **Dynamic Invoice Items Table**

The application has a dynamic table in which items may be added or deleted dynamically according to the requirement of the invoice. The fields in each row include description of goods, HSN/SAC code, quantity of bags, quantity, rate, discount, GST percentage, and the amount, which are calculated automatically. The table keeps itself updated every time any changes are made, and it makes it convenient for invoices to manage more than one item.

1. **Automated Calculations**

One of the most important aspects of this application is that it automatically calculates the total. When the user inputs the quantity, rate, discount, and GST rate percentage, the application calculates the amount for each item. The application adds all the item totals to calculate the invoice total. On top of this, a "Super Discount" can be added at the level of the total, after which the system calculates and shows the grand total. This automation prevents errors and provides precise billing.

1. **Amount in Words Conversion**

For enhanced clarity and adherence to financial documentation norms, the application translates the final invoice value into words. It is based on the Indian system of numbering (Thousand, Lakh, and Crore) rather than the international system. For instance, a figure like 1,25,000 will be displayed as "INR One Lakh Twenty Five Thousand Only," making the invoice professional as well as acceptable for business purposes.

1. **Smart Autocomplete for Frequently Used Fields**

To save effort and time, the system retains previously entered information like buyer names, addresses, GST numbers, vehicle numbers, and destinations. When the user types these fields, the application auto-suggests previously stored values, through which quick selection can be made using autocomplete. This saves repetitive typing and ensures consistency across invoices.

1. **Automatic Invoice Numbering**

The system has an invoice numbering mechanism that maintains proper sequencing of invoice records. As a new invoice is created and printed, the system automatically raises the invoice number for the subsequent transaction. It prevents duplication, maintains the records in good order, and makes it easy to track and retrieve them in the future.

1. **Persistent Data Storage**

All the invoice information is stored in the browser's storage so that users can reuse and access previous invoices without having to reinput information. The feature offers simple record-keeping functionality for easy viewing, editing, or copying invoices when needed. Persistent storage also assists with autocomplete functionality by saving common inputs.

1. **HSN/SAC Summary Section**

The invoice provides a special area to tally up HSN/SAC codes and their respective taxable amounts. This area gets automatically updated to the grand total, ensuring precise reporting. It also facilitates preparation of GST documents since HSN/SAC codes play a fundamental compliance role in India.

1. **Print-Ready Layout**

The app has been designed to create invoices in a professional, print-ready layout. Print-specific CSS is used to ensure extra elements, such as buttons and controls, are masked during printing. This allows users to print invoices directly without any additional formatting, cutting down on time and effort.

1. **Declaration and Bank Details Section**

In accordance with standard business practice, the invoice also has a statement declaring the invoice to be genuine. Company bank details are also clearly exhibited so that clients can pay through their preferred method easily. This aspect proves that invoices are not only professional but also legally compliant.

1. **User-Friendly Interface**

The overall design of the application prioritizes simplicity and usability. Fields are neatly arranged, inputs are easy to fill, and controls like “Add Row,” “Remove Row,” and “Print Invoice” provide intuitive navigation. The layout resembles a traditional invoice format, making it familiar for business users while still benefiting from automation and digital features.

## SOFTWARE REQUIREMNTS

The successful development and execution of the Invoice Application depends on the following software requirements:

1. Operating System

(The application is platform-independent as it runs on any modern browser.)

* Windows 10 or later
* Linux (Ubuntu 20.04 or above)
* macOS (Catalina or later)

1. Web Browser

(Any browser with JavaScript, HTML5, and CSS3 support is sufficient

* Google Chrome (latest version recommended)
* Mozilla Firefox
* Microsoft Edge
* Safari

1. Development Tools / Editors

(Used for coding and modifying HTML, CSS, and JavaScript files.)

* Visual Studio Code (preferred)
* Sublime Text / Atom / Notepad++ (alternative editors)

4.Programming Languages & Technologies

* HTML5 – for structuring the invoice layout
* CSS3 – for styling and creating a print-ready design
* JavaScript (ES6+) – for logic, calculations, and data handling

5.Optional Tools

* Git & GitHub (for version control and project backup)
* WPS Office / MS Word (for documentation and report preparation)
* Browser Developer Tools (for debugging and testing application behavior)

## IMPLEMENTATION

## The Invoice Application has been implemented using a combination of modern web technologies that provide a balance of simplicity, flexibility, and efficiency. Each technology plays a specific role in the system, ensuring that the application is user-friendly, reliable, and capable of handling the invoicing requirements of small and medium-scale businesses. The key technologies used are as follows:

## Technology Description:

### ****1. HTML5 (HyperText Markup Language)****

HTML5 is the backbone of the application, used to design and structure the invoice layout. All the essential components of an invoice—such as company details, buyer details, invoice number, itemized tables, HSN/SAC summary, total amounts, declaration, and bank details—are structured using HTML elements.

* **Why HTML5?**  
  HTML5 was chosen because it is lightweight, universally supported, and does not require any additional software to run. It provides semantic tags that make the structure of the invoice clear, improves readability, and ensures compatibility across all browsers. Additionally, HTML5 supports forms and input types that simplify data entry for users.

### ****2. CSS3 (Cascading Style Sheets)****

CSS3 is used to style the invoice and give it a professional, print-ready appearance. It controls aspects such as font styles, table alignment, spacing, borders, and the overall visual presentation of the invoice. Print-specific CSS rules are also applied to ensure that the invoice looks neat when printed, with unnecessary buttons or controls hidden.

* **Why CSS3?**  
  CSS3 was chosen because it allows complete control over the design and layout without affecting functionality. It ensures that the invoice generated looks professional and is aligned with standard business formats. With CSS media queries, the same design works well for both on-screen display and printing, saving time and effort for the user.

### ****3. JavaScript (ES6+)****

JavaScript powers the logic and interactivity of the application. It is responsible for dynamic calculations such as totals, discounts, GST amounts, and grand totals. JavaScript also manages features like adding/removing rows in the invoice, converting amounts into words (Indian numbering system), storing frequently used values, generating invoice numbers, and enabling autocomplete for fields.

* **Why JavaScript?**  
  JavaScript was chosen because it provides the flexibility to add dynamic behavior directly in the browser without needing additional software or external servers. Its ability to manipulate the Document Object Model (DOM) makes it ideal for building interactive features like real-time calculations and autocomplete. Being widely supported, it ensures the application can run smoothly across all major browsers.

### ****4. Browser Local Storage****

The application uses the localStorage feature of web browsers to save user inputs, past invoice data, and frequently used fields. This ensures persistence of data even when the page is refreshed or reopened later.

* **Why Local Storage?**  
  Local storage was chosen because it allows data to be stored on the user’s device without requiring an internet connection or database setup. It is secure, fast, and sufficient for small-scale business needs where data volumes are manageable.

### ****5. Print-Specific Technologies****

The implementation includes print-optimized styles and layouts to allow invoices to be generated in a ready-to-print format. This avoids the need for users to manually adjust formatting before printing.

* **Why Print Optimization?**  
  Print-ready formatting was chosen to make invoice generation quick and hassle-free. By ensuring the output is professional, clean, and business-compliant, the application saves users time and guarantees that invoices are consistent in appearance.
* **CODE USED TO DEVELOP THIS APPLICATION:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Bill of Supply (Frontend Only)</title>

<style>

body { font-family: Arial, sans-serif; background: #f7f7f7; margin: 0; }

.sheet { width: 210mm; min-height: 297mm; margin: 12px auto; background: #fff; padding: 10mm; box-shadow: 0 0 0.5mm rgba(0,0,0,.3); }

table { width: 100%; border-collapse: collapse; table-layout: fixed; }

th, td { border: 1px solid #000; padding: 4px 6px; vertical-align: top; font-size: 12px; }

.noborder td, .noborder th { border: none; }

.center { text-align: center; } .right { text-align: right; } .bold { font-weight: 700; }

.field { width: 100%; border: none; outline: none; font: inherit; background: transparent; }

.field.inline { display: inline-block; min-width: 80px; border-bottom: 1px dashed #aaa; }

.field.right { text-align: right; }

.controls { margin: 6px 0; text-align: right; }

.controls button, .print-btn { padding: 4px 10px; cursor: pointer; border: 1px solid #000; background: #fafafa; font-size: 12px; }

.print-btn { margin-top: 8px; }

.totals-label { padding-left: 60%; font-weight: bold; }

.autocomplete-suggestions { position: absolute; background: #fff; border: 1px solid #ccc; max-height: 140px; overflow-y: auto; font-size: 12px; z-index: 1000; display: none; }

.autocomplete-suggestions div { padding: 4px; cursor: pointer; }

.autocomplete-suggestions div.active, .autocomplete-suggestions div:hover { background: #eee; }

.autocomplete-wrapper { position: relative; display: inline-block; width: 100%; }

@media print { .no-print { display: none; } .sheet { margin: 0; box-shadow: none; } .field { border: none; } }

</style>

</head>

<body>

<div class="sheet" id="sheet">

<!-- HEADER -->

<div class="center bold" style="margin-bottom:4px;">Bill of Supply (ORIGINAL FOR RECIPIENT)</div>

<table>

<tr>

<!-- Left: Company and Buyer -->

<td style="width:50%">

<div class="bold">ABC (CHENNAI)</div>

<div>NO:25, 3B 3RD FLOOR, TNAGAR, CHENNAI600017, TAMILNADU, INDIA</div>

<div>GST: <b>12234567899</b></div>

<hr>

<div class="bold">Buyer (Bill to)</div>

<div class="autocomplete-wrapper">

<input class="field bold" id="buyerName" placeholder="DMART" />

<div class="autocomplete-suggestions" id="suggest\_buyerName"></div>

</div>

<div class="autocomplete-wrapper">

<textarea class="field" rows="2" id="buyerAddress" placeholder="#390, T.H. Road, Thiruvottiyur, Chennai - 600019"></textarea>

<div class="autocomplete-suggestions" id="suggest\_buyerAddress"></div>

</div>

<div>GST: <div class="autocomplete-wrapper"><input class="field inline" id="buyerGST" /><div class="autocomplete-suggestions" id="suggest\_buyerGST"></div></div></div>

<div>Email: <input class="field inline" id="buyerEmail" /></div>

<div>FSSAI No.: <input class="field inline" id="buyerFSSAI" /></div>

</td>

<!-- Right: Invoice Metadata -->

<td style="width:50%">

<table style="width:100%; border:none;">

<tr><td>Invoice No.</td>

<td colspan="3">

<div class="autocomplete-wrapper">

<input class="field bold" id="invoiceNo" />

<div class="autocomplete-suggestions" id="suggest\_invoiceNo"></div>

</div>

</td>

</tr>

<tr><td>Dated</td><td colspan="3"><input class="field inline" id="date" placeholder="29-Aug-25" /></td></tr>

<tr><td>Delivery Note</td><td><input class="field inline" id="deliveryNote" /></td><td>Mode/Terms of Payment</td><td><input class="field inline" id="termsPayment" /></td></tr>

<tr><td>Reference No. & Date</td><td colspan="3"><input class="field inline" id="refNoDate" /></td></tr>

<tr><td>Other References</td><td colspan="3"><input class="field inline" id="otherRef" /></td></tr>

<tr><td>Buyer’s Order No.</td><td><input class="field inline" id="buyerOrderNo" /></td><td>Dated</td><td><input class="field inline" id="buyerOrderDate" /></td></tr>

<tr><td>Dispatch Doc No</td><td><input class="field inline" id="dispatchDocNo" /></td><td>Delivery Note Date</td><td><input class="field inline" id="deliveryNoteDate" /></td></tr>

<tr><td>Dispatched through</td><td><div class="autocomplete-wrapper"><input class="field inline" id="dispatchedThrough" /><div class="autocomplete-suggestions" id="suggest\_dispatchedThrough"></div></div></td><td>Destination</td><td><div class="autocomplete-wrapper"><input class="field inline" id="destination" /><div class="autocomplete-suggestions" id="suggest\_destination"></div></div></td></tr>

<tr><td>Bill of Lading / LR-RR No.</td><td><input class="field inline" id="lrrr" /></td><td>Motor Vehicle No.</td><td><div class="autocomplete-wrapper"><input class="field inline" id="vehicle" placeholder="TN 18 S 2224" /><div class="autocomplete-suggestions" id="suggest\_vehicle"></div></div></td></tr>

<tr><td colspan="4">Terms of Delivery <input class="field inline" id="terms" placeholder="DELIVERY AT ..." /></td></tr>

</table>

</td>

</tr>

</table>

<!-- ITEMS -->

<table id="itemsTable">

<thead>

<tr>

<th>Sl No</th>

<th>Description of Goods</th>

<th>HSN/SAC</th>

<th>No of Bags</th>

<th>Quantity (Kg)</th>

<th>Rate (Per Bag)</th>

<th>Discount</th>

<th>GST%</th>

<th>Amount</th>

</tr>

</thead>

<tbody></tbody>

</table>

<div class="controls no-print">

<button onclick="addRow()">+ Add Row</button>

<button onclick="removeRow()">− Remove Last Row</button>

</div>

<!-- TOTALS -->

<table>

<tr>

<td style="width:50%">ACK Date <input class="field inline" id="ackDate" /></td>

<td class="right"></td>

</tr>

<tr>

<td class="totals-label">Total</td>

<td class="right"><input class="field right inline" id="total" /></td>

</tr>

<tr>

<td class="totals-label">Super Discount</td>

<td class="right">

<input class="field right inline" id="superDiscount" /> %

</td>

</tr>

<tr>

<td class="totals-label">Grand Total</td>

<td class="right"><input class="field right inline" id="grandTotal" /></td>

</tr>

<tr>

<td colspan="2" class="bold">

Amount Chargeable (in words): <input class="field inline" id="amountWords" />

</td>

</tr>

</table>

<!-- HSN Summary -->

<table>

<thead><tr><th>HSN/SAC</th><th>Taxable Value</th></tr></thead>

<tbody>

<tr><td><input class="field" id="hsnSummary" placeholder="07139010"/></td><td><input class="field right" id="hsnValue" placeholder="9,87,945.00"/></td></tr>

<tr><td class="right bold">Total</td><td><input class="field right" id="hsnTotal" placeholder="9,87,945.00"/></td></tr>

</tbody>

</table>

<div>Tax Amount (in words): <b id="taxAmountWords">NIL</b></div>

<!-- Bank + Declaration + Receiver -->

<table>

<tr>

<td style="width:50%; vertical-align:top;">

<div class="bold">Remarks:</div>

<input class="field" id="remarks" />

<div class="bold">Declaration</div>

<textarea class="field" rows="2" id="declaration">We declare that this invoice shows the actual price of the goods described and that all particulars are true</textarea>

</td>

<td style="width:50%; vertical-align:top;">

<div class="bold">Company’s Bank Details</div>

<div>A/c Holder’s Name : <b>ABC</b></div>

<div>Bank Name A/c No. : <b>STATE BANK OF INDIA - 4340733</b></div>

<div>Branch & IFS Code : <b>SBIN0001234, BESANT NAGAR</b></div>

<br>

<div>Receiver name:</div>

<div class="field" style="height:8mm"></div>

<div>Receiver Signature:</div>

<div class="field" style="height:8mm"></div>

<div>Receiver seal:</div>

<div class="field" style="height:8mm"></div>

<div>Kindly acknowledge the goods are in good condition</div>

<div style="text-align:right; margin-top:10px;">FOR <b>ABC (CHENNAI)</b></div>

<div style="text-align:right;">Authorised Signatory</div>

</td>

</tr>

</table>

<!-- Print Button -->

<div class="no-print" style="text-align:center; margin-top:10px;">

<button class="print-btn" onclick="saveAndPrint()">🖨️ Print Invoice</button>

</div>

</div>

<script>

// ========= Helpers =========

const $ = (sel, root=document) => root.querySelector(sel);

const $$ = (sel, root=document) => Array.from(root.querySelectorAll(sel));

const itemsBody = $('#itemsTable tbody');

function parseNum(v){ return parseFloat((v||'').toString().replace(/,/g,''))||0; }

function formatNum(n){ return (Number.isFinite(n)?n:0).toLocaleString('en-IN',{minimumFractionDigits:2, maximumFractionDigits:2}); }

// ========= Initial Invoice No =========

let currentInvoiceNo = localStorage.getItem('lastInvoiceNo') || '1B1001';

$('#invoiceNo').value = currentInvoiceNo;

// ========= Items table =========

function addRow(data={}) {

const i = itemsBody.children.length;

const tr = document.createElement('tr');

tr.innerHTML = `<td class='center'>${i+1}</td>

<td><input class='field' data-k='desc' value='${data.desc||''}'></td>

<td><input class='field center' data-k='hsn' value='${data.hsn||''}'></td>

<td><input class='field center' data-k='bags' value='${data.bags||''}'></td>

<td><input class='field center' data-k='qty' value='${data.qty||''}'></td>

<td><input class='field center' data-k='rate' value='${data.rate||''}'></td>

<td><input class='field center' data-k='discount' value='${data.discount||''}'></td>

<td><input class='field center' data-k='gst' value='${data.gst||''}'></td>

<td><input class='field right' data-k='amount' value='${data.amount||''}' readonly></td>`;

itemsBody.appendChild(tr);

tr.querySelectorAll('input').forEach(inp => inp.addEventListener('input', recalc));

recalc();

}

function removeRow(){ if(itemsBody.children.length){ itemsBody.removeChild(itemsBody.lastChild); reindex(); recalc(); }}

function reindex(){ $$('#itemsTable tbody tr').forEach((tr,i)=> tr.firstElementChild.textContent=i+1); }

function recalc(){

let total=0;

$$('#itemsTable tbody tr').forEach(tr=>{

const g=k=>tr.querySelector(`[data-k=${k}]`).value;

let qty=parseNum(g('qty')), rate=parseNum(g('rate')),

disc=parseNum(g('discount')), gst=parseNum(g('gst'));

let amt = qty\*rate - disc; // discount before GST

amt += amt\*gst/100;

tr.querySelector('[data-k=amount]').value = formatNum(amt);

total += amt;

});

$('#total').value = formatNum(total);

// Super Discount as percentage

const sdPercent = parseNum($('#superDiscount').value);

const sdAmount = total \* (sdPercent / 100);

const grand = total - sdAmount;

$('#grandTotal').value = formatNum(grand);

// Words

const words = numberToWords(grand);

$('#amountWords').value = words;

// ✅ Sync HSN values with Grand Total

$('#hsnValue').value = formatNum(grand);

$('#hsnTotal').value = formatNum(grand);

// ✅ Sync Tax Amount (in words) with Amount Chargeable (in words)

$('#taxAmountWords').textContent = words;

}

function numberToWords(num){

num = Math.floor(num||0);

if(num===0) return 'INR Zero Only';

const a=['','One','Two','Three','Four','Five','Six','Seven','Eight','Nine','Ten','Eleven','Twelve','Thirteen','Fourteen','Fifteen','Sixteen','Seventeen','Eighteen','Nineteen'];

const b=['','','Twenty','Thirty','Forty','Fifty','Sixty','Seventy','Eighty','Ninety'];

function inWords(n){

if(n<20) return a[n];

if(n<100) return (b[Math.floor(n/10)] + (n%10?' '+a[n%10]:''));

if(n<1000) return (a[Math.floor(n/100)]+' Hundred'+(n%100?' '+inWords(n%100):''));

if(n<100000) return (inWords(Math.floor(n/1000))+' Thousand'+(n%1000?' '+inWords(n%1000):''));

if(n<10000000) return (inWords(Math.floor(n/100000))+' Lakh'+(n%100000?' '+inWords(n%100000):''));

return (inWords(Math.floor(n/10000000))+' Crore'+(n%10000000?' '+inWords(n%10000000):''));

}

return 'INR '+inWords(num)+' Only';

}

// ========= Form data IO =========

function getFormData(){

const data={};

$$('.field').forEach(f=>{ if(f.id) data[f.id]=f.value; });

data.items = $$('#itemsTable tbody tr').map(tr=>{

const obj={};

tr.querySelectorAll('input').forEach(inp=>{ obj[inp.dataset.k]=inp.value; });

return obj;

});

return data;

}

function setFormData(data){

$$('.field').forEach(f=>{ if(f.id && data[f.id]!==undefined) f.value=data[f.id]; });

// items

itemsBody.innerHTML='';

if(!data.items || data.items.length===0){ addRow(); } else { data.items.forEach(it=>addRow(it)); }

recalc();

}

function clearForm(){

$$('.field').forEach(f=>{ if(f.id!=='invoiceNo') f.value=''; });

itemsBody.innerHTML='';

addRow();

}

// ========= Invoice number helpers =========

function incrementInvoiceNo(no){

const prefix = (no.match(/^[^0-9]\*/)||[''])[0];

const digits = no.replace(/\D/g,'');

const n = parseInt(digits||'0',10)+1;

return prefix + n.toString().padStart(digits.length||4,'0');

}

// ========= localStorage keys =========

const INV\_PREFIX = 'invoice\_';

function saveInvoice(no, data){ localStorage.setItem(INV\_PREFIX+no, JSON.stringify(data)); }

function loadInvoice(no){ const raw=localStorage.getItem(INV\_PREFIX+no); return raw?JSON.parse(raw):null; }

// ========= Autocomplete (with keyboard nav) =========

function addToHistory(field, value){

if(!value) return; const key='history\_'+field;

let arr=JSON.parse(localStorage.getItem(key)||'[]');

if(!arr.includes(value)) arr.unshift(value);

if(arr.length>200) arr=arr.slice(0,200);

localStorage.setItem(key, JSON.stringify(arr));

}

function getHistory(field){ return JSON.parse(localStorage.getItem('history\_'+field)||'[]'); }

function bindAutocomplete(inputId){

const inp = $('#'+inputId);

const box = $('#suggest\_'+inputId);

let idx = -1; // active index for keyboard

function close(){ box.style.display='none'; idx=-1; $$('.active', box).forEach(n=>n.classList.remove('active')); }

function open(){ box.style.display = box.children.length? 'block':'none'; }

function build(){

const q = inp.value.trim().toLowerCase();

const list = getHistory(inputId).filter(v=>!q || v.toLowerCase().includes(q));

box.innerHTML = list.map(v=>`<div>${v.replace(/</g,'&lt;')}</div>`).join('');

idx=-1; open();

$$('#suggest\_'+inputId+' div').forEach((div,i)=>{

div.addEventListener('mousedown', e=>{ e.preventDefault(); select(i); });

});

}

function select(i){

const el = box.children[i]; if(!el) return; inp.value = el.textContent; close(); inp.dispatchEvent(new Event('input')); inp.blur(); inp.focus();

}

inp.addEventListener('input', ()=>{

// special case: invoiceNo exact match autoload

if(inputId==='invoiceNo'){

const ex = loadInvoice(inp.value.trim());

if(ex) setFormData(ex);

}

build();

});

inp.addEventListener('keydown', e=>{

if(box.style.display!=='block') return;

const max = box.children.length-1;

if(e.key==='ArrowDown'){ e.preventDefault(); idx = Math.min(max, idx+1); highlight(); }

else if(e.key==='ArrowUp'){ e.preventDefault(); idx = Math.max(0, idx-1); highlight(); }

else if(e.key==='Enter'){ if(idx>-1){ e.preventDefault(); select(idx);} }

else if(e.key==='Escape'){ close(); }

});

function highlight(){ $$('#suggest\_'+inputId+' div').forEach((d,i)=> d.classList.toggle('active', i===idx)); }

inp.addEventListener('blur', ()=> setTimeout(close, 120));

}

// Bind all autocompletes

['invoiceNo','buyerName','buyerAddress','buyerGST','dispatchedThrough','destination','vehicle',]

.forEach(bindAutocomplete);

// ========= Save & Print =========

function saveAndPrint(){

const no = $('#invoiceNo').value.trim();

if(!no){ alert('Invoice number is required'); return; }

const data = getFormData();

// save main histories

['invoiceNo','buyerName','buyerAddress','buyerGST','dispatchedThrough','destination','vehicle']

.forEach(f=> addToHistory(f, data[f]||''));

saveInvoice(no, data);

window.print();

// increment invoice number and reset

const next = incrementInvoiceNo(no);

localStorage.setItem('lastInvoiceNo', next);

$('#invoiceNo').value = next;

clearForm();

}

// ========= Auto-load last invoice on revisit =========

// Start with a clean row visible

addRow();

// If user types an existing invoice number, load instantly (handled in autocomplete input listener)

// Recalculate when totals inputs change manually

['superDiscount'].forEach(id=> $('#'+id).addEventListener('input', recalc));

</script>

</body>

</html>

## OUTPUT SCREENS

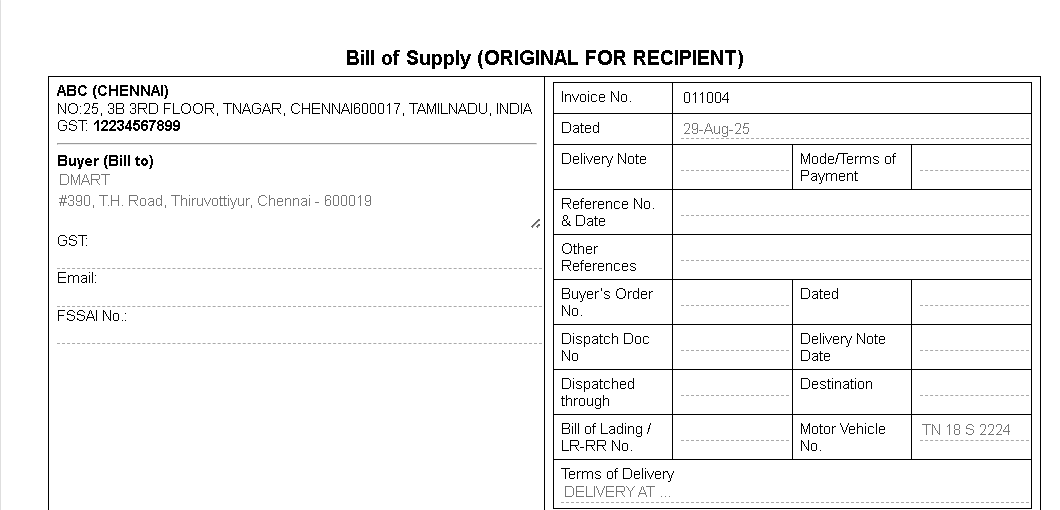


Fig: 6.1

* This Bill of Supply format shown includes two main sections. The left side captures the seller and buyer details, such as business name, address, GSTIN, and FSSAI number. The right side records the invoice details, including invoice number, date, references, dispatch information, vehicle number, and payment terms. This structured format ensures all essential transaction details are clearly presented for record-keeping and compliance.

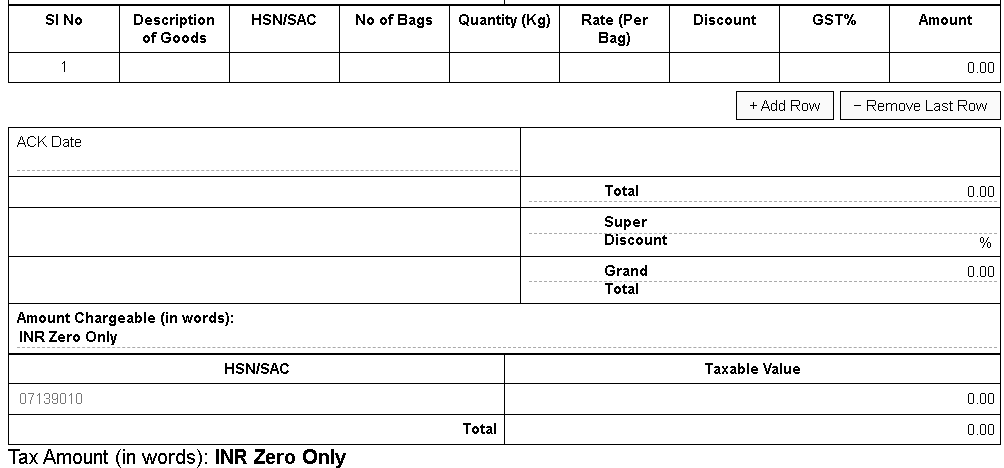


Fig: 6.2

* This section of the Bill of Supply displays the itemized goods table with details such as description, HSN/SAC code, quantity, rate, discount, GST, and calculated amount. Below it, the totals panel summarizes the overall total, super discount, and grand total, along with the amount chargeable in words. It also includes the HSN summary table, ensuring clarity in tax reporting and compliance.

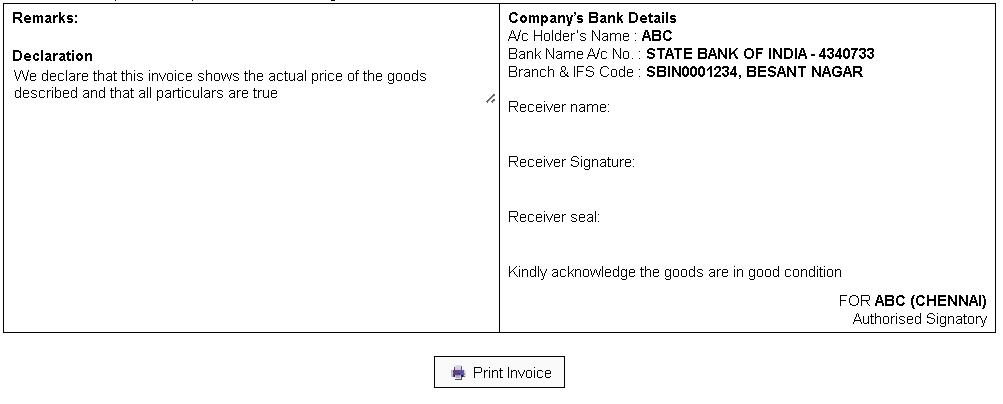


Fig: 6.3

* This section contains the **remarks and declaration** ensuring the authenticity of the invoice details, alongside the **company’s bank details** for payments. It also provides space for the **receiver’s name, signature, and seal** to confirm receipt of goods. At the bottom, a **print button** is available to generate a hard copy of the invoice.

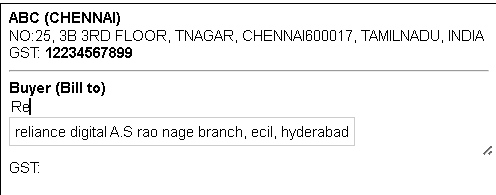


Fig: 6.4

* The application includes an **autocomplete feature** for buyer details. It stores previously entered values in local storage and, when the user types, shows matching suggestions similar to a search engine. This saves time, avoids errors, and makes data entry faster and more convenient.

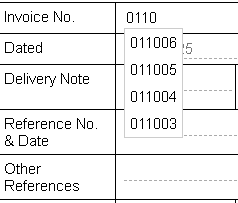


Fig: 6.5

* The application **auto-increments the invoice number** after each print, ensuring continuity in billing records. It also provides access to **previously used invoices**, allowing users to quickly load, update, or edit the related data as needed.

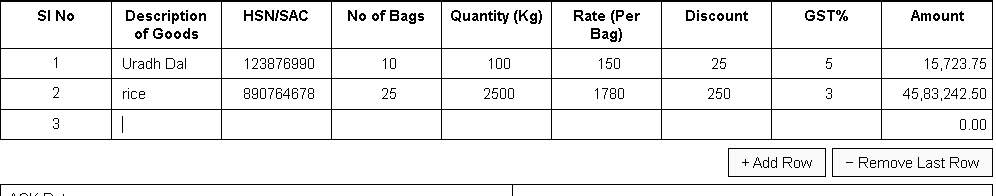


Fig: 6.6

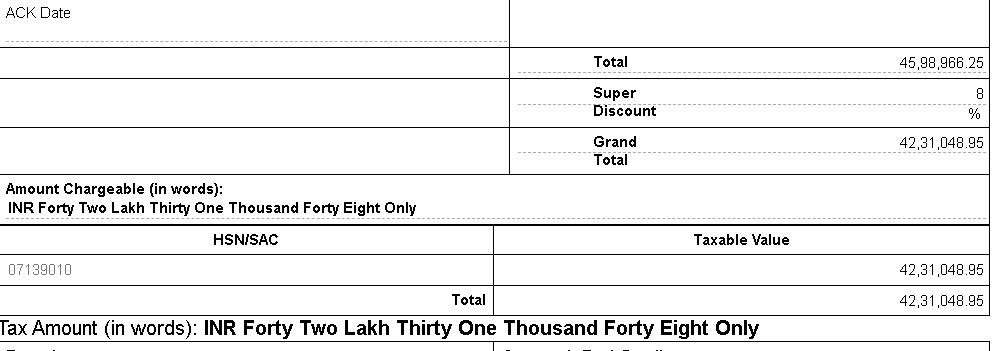
* The application provides an “Add Row” option to enter details of additional goods for calculation, along with a “Remove last Row” option to delete unnecessary entries. This makes the invoice flexible and adaptable to varying numbers of items.

Fig: 6.7

* The application performs automatic calculations of total value, super discount, grand total, and taxable value, while also converting the final amount into words. This ensures accuracy, saves time, and produces invoices in a professional format.

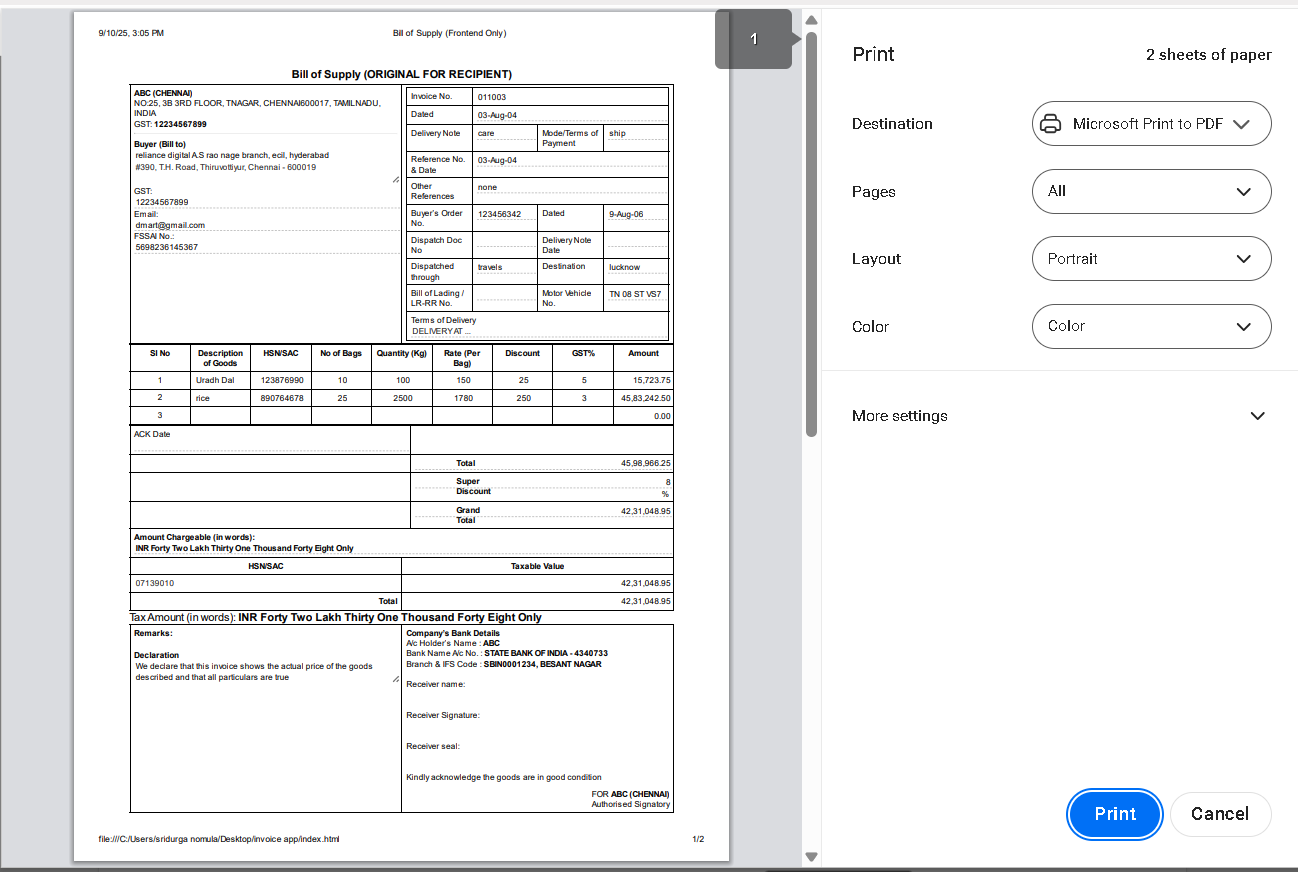


Fig: 6.8

* Provides print option that enables us to print the invoice edited or stored into pdf format in our respective system.

## CONCLUSION

The Invoice Application built in this project offers a dependable, efficient, and user-friendly means of creating expert invoices. It solves the prevalent problems of manual billing like taking up a lot of time, calculation mistakes, and redundant data input by bringing automation into the process. The system makes sure there is correct calculation of totals, discounts, GST amounts, and even the final sum is converted to words, making the invoices precise and professional. Functions like autocomplete for customer information, auto-incrementing invoice numbers, and the functionality to load and edit generated invoices provide a boost in terms of speed, adaptability, and documentation. The Add Row and Remove Row functionalities allow the application to work with invoices that have an unspecified number of items, while the HSN summary and taxable value fields facilitate with taxation standards. The integrated print functionality enables invoices to be automatically printed in a professional layout. As the application is developed with lightweight technologies such as HTML, CSS, and JavaScript, it is easy, accessible, and does not need additional software or complicated setup, which makes it very applicable to small and medium-scale enterprises. With less human error and time saved, the system enhances trustworthiness, efficiency, and reliability in business practices. Its clean but professional layout makes it possible for even the less technologically savvy users to effectively use it. The project showcases how digital technology can make optimum use of everyday processes and emphasizes the relevance of automation in contemporary financial management. Overall, the Invoice Application achieves its aim of providing an intelligent, precise, and professional billing application, and also shows that small digital solutions can make a huge difference in streamlining daily business processes.