

No Budget, No Problem: Building Custom AI Tools To Improve AP Efficiency

-  Automate More
-  Save Without Spending
-  Improve Accuracy
-  Save Time

SMARTER PROCESS. STRONGER RESULTS.



Presented by: Martin Girard, CPA

Do you need NASBA CPE credits?

- Navigate to website: iofm.cnf.io or scan the QR code →
- Check-in and check-out of your sessions to track your attendance for NASBA CPEs
- Certified with IOFM? No need to check-in and out of sessions. Self-report CEUs on IOFM.com instead after the event!



About me

CPA with nearly 25 years of experience in finance & accounting

- 15 years at Barrette Structural
- 10 years across multiple organizations

Specializations:

- Data analysis (Power BI, Access)
- Process improvement
- Credit management

Passionate about **turning data & AI** into **practical tools**

Active speaker in Credit & AP communities across Canada & U.S.



IOFM Spring

CONFERENCE & EXPO

 Tue May 12th, 2026  3:00 PM






No Budget, No Problem:
Building Custom AI Tools
To Improve AP Efficiency

May 2026 | Loews Sapphire Falls Resort | Universal Orlando, FL

IOFM Spring
CONFERENCE & EXPO

The Reality of Finance Work

-  **We don't lack data...**
We lack **time** to use it properly.
-  **We don't lack systems...**
We struggle to **connect** them.
-  **We don't lack talent...**
We waste it on **repetitive work**.



TIME
is the
constraint



AI
is **NOT**
the solution

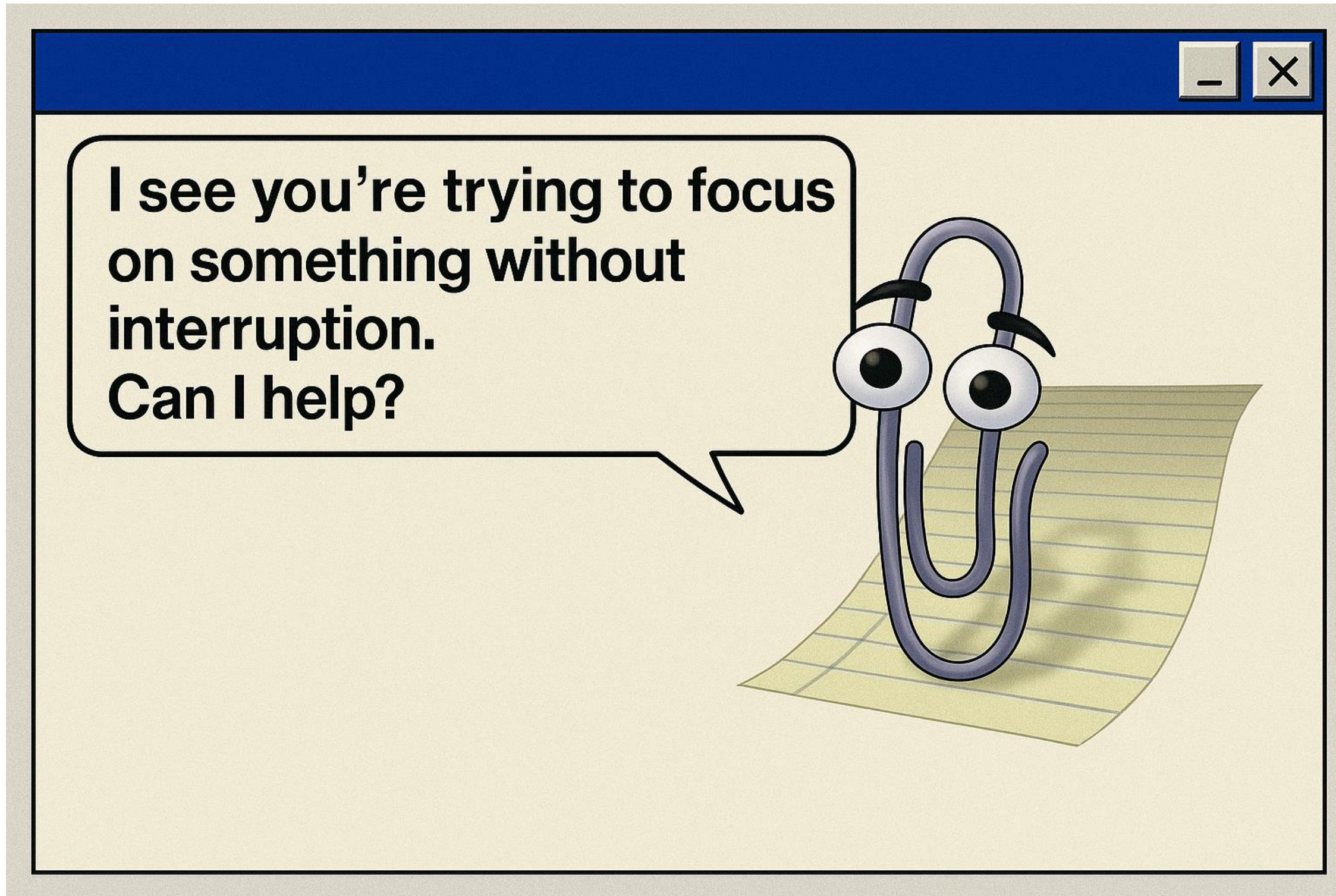


The goal is simple: **Free time**. Focus on what drives value.

My Formula for Success



- ✓ Respect people
- ✓ Take responsibility
- ✓ Be proactive
- ✓ Keep it simple
- ✓ Focus on high-value work
- ✓ Results over excuses



DEMO 1

LinkedIn Profile → Business Application



LinkedIn
Profile Data



ChatGPT
AI Analysis

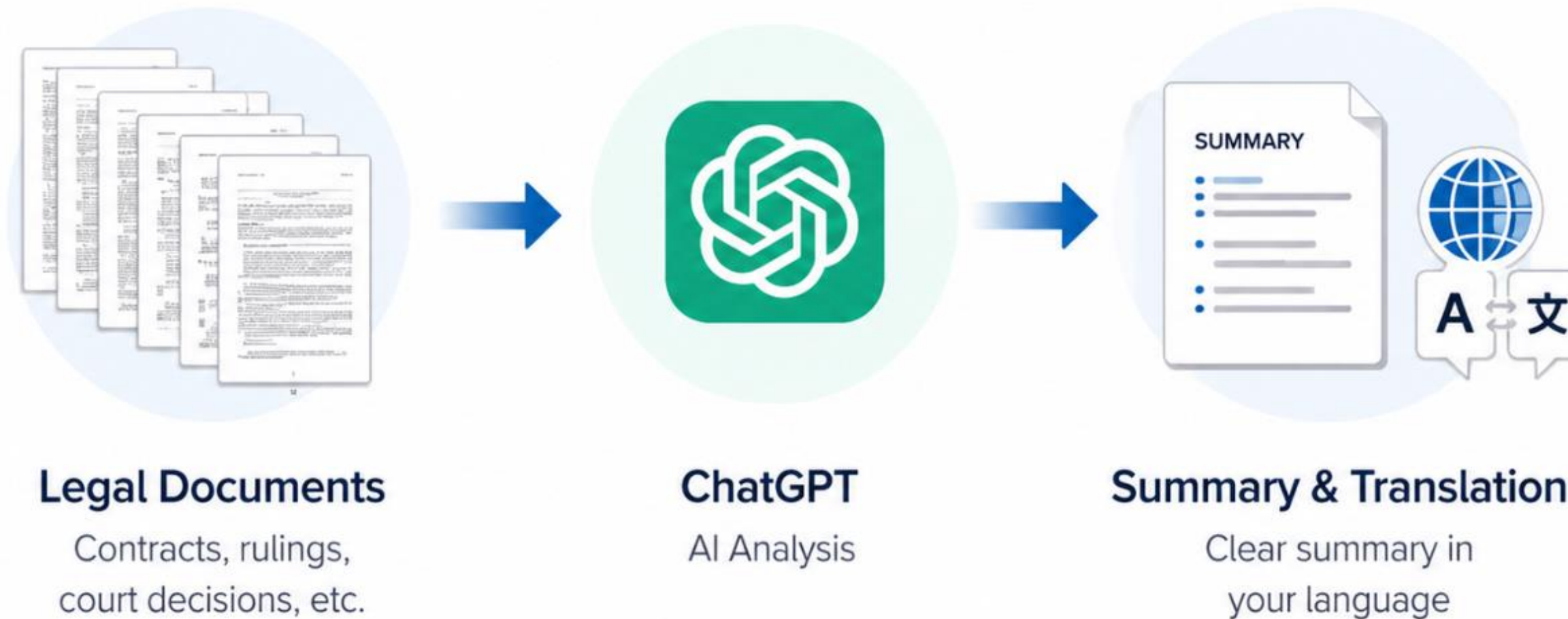


Business Application
Insights & Action

DEMO 2

Summarize and translate a legal decision

From complex documents to clear understanding



DEMO 3

Build a Cash Application Tool in Minutes

From idea to working tool: fast and easy



DEMO 3



Build a Cash Application Tool in Minutes

I have a payment of 37635.65

The open invoices are:

Invoice Amount

90454489 11 476.44

90457552 531.91

90459191 20 399.10

90459436 8 781.03

90459437 400.87

90459438 977.62

90459439 14 161.19

90459720 32 543.79

90460426 1 458.74

90460921 3 789.88

90461103 409.60

90461446 321.93

90461448 1 301.98

90461838 779.88

90461925 1 797.87

90462296 8 920.15

90462522 24 191.11

90462524 7 238.68

90462618 1 431.81

90462791 2 238.13

what invoices are paid?

Analyzed ▾

The invoices that add up exactly to your payment of 37,635.65 are:

- 90459720 → 32,543.79
- 90460921 → 3,789.88
- 90461448 → 1,301.98

✓ Total = 37,635.65

DEMO 3

Build a Cash Application Tool in Minutes

please create an ultra simple html code so I would have a cash application tool

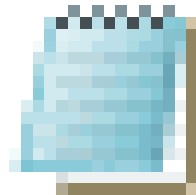
Thought for 16s

```
html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Ultra-Simple Cash Application</title>
<style>
body{font-family:system-ui,Arial,Helvetica,sans-serif;max-width:820px;margin:24px auto;
hi{font-size:18px;margin:0 0 12px 0}
.row{display:flex;gap:12px;align-items:center;flex-wrap:wrap;margin-bottom:10px}
input,textarea,button{font:inherit;border:1px solid #ccc;border-radius:8px;padding:8px}
input{width:180px}
textarea{width:100%;height:210px;resize:vertical}
button{cursor:pointer}
.out{white-space:pre-wrap;border:1px solid #5e5e5;border-radius:8px;padding:12px}
small{color:#666}
</style>
</head>
<body>
<h1>Cash Application (Ultra-Simple)</h1>
<div class="row">
<label>Payment amount:
<input id="payment" placeholder="e.g. 37 635.65" />
</label>
<button id="run">Find Invoices</button>
<button id="clear" title="Clear inputs">Clear</button>
</div>
<textarea id="list" placeholder="Paste your open invoices here, one per line.
Accepted formats (any mix of spaces, tabs):
98454489 11 476.44
98457552 531.91
98459720 32 543.79 x (the trailing x is ignored)
"></textarea>
<div class="out" id="out"><small>Result will appear here.</small></div>
<script>
function toCents(s){
// keep digits and decimal point; drop spaces and commas
s = String(s).replace(/[^d.,]/g,'').replace(/,/g,'').replace(/./g,'');
if(!s) return NaN;
// ensure decimal with dot
if(!s.includes('.') || s.endsWith('0')) s += '.00';
return Math.round(parseFloat(s) * 100);
}
function parseLines(text){
const rows = [];
for(const raw of text.split(/\r?\n/)){
const line = raw.trim();
if(!line) continue;
// invoice: first 6+ consecutive digits
const invMatch = line.match(/^(b\d{6,})b/);
// amount: last numeric-looking token
const amtMatches = line.match(/[\ds.]+(?:\.\d{1,2})?/g);
const amtToken = amtMatches ? amtMatches[amtMatches.length-1] : null;
if(!invMatch || !amtToken) continue;
const inv = invMatch[0];
const cents = toCents(amtToken);
if(Number.isFinite(cents)) rows.push([inv, cents]);
}
return rows;
}
function subsetSum(items, targetCents, maxDepth=7){
// DFS with pruning; items sorted by amount desc
items.sort((a,b)->b.cents-a.cents);
const n = items.length;
let best = {diff: Infinity, pick: []};
function dfs(i, depth, sum, pick){
const diff = Math.abs(targetCents - sum);
if(diff < best.diff) best = {diff, pick: pick.slice()};
if(sum === targetCents) throw {pick: pick.slice()}; // fast exit on exact match
if(depth === 0 || i === n || sum > targetCents + best.diff) return;
// optimistic bound: if even adding all remaining can't improve, stop
for(let k=i;k<n;k++){
if(depth<0) break;
pick.push(items[k]);
dfs(k+1, depth-1, sum + items[k].cents, pick);
pick.pop();
}
}
for(let depth=1; depth<=maxDepth; depth++){
try{ dfs(0, depth, 0, []); }catch(e){ return {exact:true, pick:e.pick}; }
}
return {exact:false, pick:best.pick};
}
function formatMoney(cents){
return (cents/100).toLocaleString(undefined,{minimumFractionDigits:2, maximumFr
}
document.getElementById('run').onclick = () => {
const pay = toCents(document.getElementById('payment').value);
const rows = parseLines(document.getElementById('list').value);
const out = document.getElementById('out');
if(!Number.isFinite(pay)) return out.textContent = 'Enter a valid payment amount';
if(rows.length === 0) return out.textContent = 'No valid invoice lines detected';
const res = subsetSum(rows, pay, 8); // search up to 8 invoices
const sum = res.pick.reduce((a,r)->a+r.cents,0);
const lines = [];
lines.push('Payment: $(formatMoney(pay))');
lines.push('Matched total: $(formatMoney(sum)) $(res.exact ? '(exact match)' : ''');
res.pick.forEach(r => lines.push(`${r.inv} - $(formatMoney(r.cents))`));
const remaining = pay - sum;
if(remaining !== 0) lines.push(`\nDifference: $(formatMoney(remaining)) $(rema
out.textContent = lines.join('\n');
};
document.getElementById('clear').onclick = () => {
document.getElementById('payment').value = '';
document.getElementById('list').value = '';
document.getElementById('out').innerHTML = '<small>Result will appear here.</small>';
};
</script>
</body>
</html>
```

```
function parseLines(text){
const rows = [];
for(const raw of text.split(/\r?\n/)){
const line = raw.trim();
if(!line) continue;
// invoice: first 6+ consecutive digits
const invMatch = line.match(/^(b\d{6,})b/);
// amount: last numeric-looking token
const amtMatches = line.match(/[\ds.]+(?:\.\d{1,2})?/g);
const amtToken = amtMatches ? amtMatches[amtMatches.length-1] : null;
if(!invMatch || !amtToken) continue;
const inv = invMatch[0];
const cents = toCents(amtToken);
if(Number.isFinite(cents)) rows.push([inv, cents]);
}
return rows;
}
function subsetSum(items, targetCents, maxDepth=7){
// DFS with pruning; items sorted by amount desc
items.sort((a,b)->b.cents-a.cents);
const n = items.length;
let best = {diff: Infinity, pick: []};
function dfs(i, depth, sum, pick){
const diff = Math.abs(targetCents - sum);
if(diff < best.diff) best = {diff, pick: pick.slice()};
if(sum === targetCents) throw {pick: pick.slice()}; // fast exit on exact mat
if(depth === 0 || i === n || sum > targetCents + best.diff) return;
// optimistic bound: if even adding all remaining can't improve, stop
for(let k=i;k<n;k++){
if(depth<0) break;
pick.push(items[k]);
dfs(k+1, depth-1, sum + items[k].cents, pick);
pick.pop();
}
}
for(let depth=1; depth<=maxDepth; depth++){
try{ dfs(0, depth, 0, []); }catch(e){ return {exact:true, pick:e.pick}; }
}
return {exact:false, pick:best.pick};
}
function formatMoney(cents){
return (cents/100).toLocaleString(undefined,{minimumFractionDigits:2, maximumFr
}
document.getElementById('run').onclick = () => {
const pay = toCents(document.getElementById('payment').value);
const rows = parseLines(document.getElementById('list').value);
const out = document.getElementById('out');
if(!Number.isFinite(pay)) return out.textContent = 'Enter a valid payment amount';
if(rows.length === 0) return out.textContent = 'No valid invoice lines detected';
const res = subsetSum(rows, pay, 8); // search up to 8 invoices
const sum = res.pick.reduce((a,r)->a+r.cents,0);
const lines = [];
lines.push('Payment: $(formatMoney(pay))');
lines.push('Matched total: $(formatMoney(sum)) $(res.exact ? '(exact match)' : ''');
res.pick.forEach(r => lines.push(`${r.inv} - $(formatMoney(r.cents))`));
const remaining = pay - sum;
if(remaining !== 0) lines.push(`\nDifference: $(formatMoney(remaining)) $(rema
out.textContent = lines.join('\n');
};
document.getElementById('clear').onclick = () => {
document.getElementById('payment').value = '';
document.getElementById('list').value = '';
document.getElementById('out').innerHTML = '<small>Result will appear here.</small>';
};
</script>
</body>
</html>
```

DEMO 3

Build a Cash Application Tool in Minutes



```
Cash app tool.html - Bloc-notes
Fichier Edition Format Affichage Aide
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Ultra-Simple Cash Application</title>
<style>
  body{font-family:system-ui,Arial,Helvetica,sans-serif;max-width:820px;marg
  h1{font-size:18px;margin:0 0 12px}
  .row{display:flex;gap:12px;align-items:center;flex-wrap:wrap;margin-bottom
  input,textarea,button{font:inherit;border:1px solid #ccc;border-radius:8px
  input{width:180px}
  textarea{width:100%;height:210px;resize:vertical}
  button{cursor:pointer}
  .out{white-space:pre-wrap;border:1px solid #e5e5e5;border-radius:8px;paddi
  < >
```

DEMO 3

Build a Cash Application Tool in Minutes

Ultra-Simple Cash Application

C:/Users/magirard/OneDrive%20-%20Barvi%20In...

Cash Application (Ultra-Simple)

Payment amount:

Paste your open invoices here, one per line.
Accepted formats (any mix of spaces, tabs):
90454489 11 476.44
90457552 531.91
90459720 32 543.79 x (the trailing x is ignored)

Result will appear here.

Now make it look very professional an futurist.

Add a logo top left. File is logo.jpg on the same folder as the html file.

Add a button for English / French / Romanian languages

Add a tolerance box

Add a copy button to copy the invoices from the result

Add designer credit

separate invoice and amount into two boxes

DEMO 3

Build a Cash Application Tool in Minutes

The screenshot shows a web application interface for 'IOFM Cash Application — Pro'. At the top right, there are language selection buttons for EN, FR, and RO. Below the header, there are two input fields: 'Payment' with the value '37 635.65' and 'Tolerance' with the value '0.00'. To the right of these fields are three buttons: 'Find' (highlighted in blue), 'Clear', and 'Copy'. The main area is divided into two columns: 'Invoices' and 'Amounts'. The 'Invoices' column contains the text: 'One invoice per line', '90454489', and '90457552'. The 'Amounts' column contains the text: 'One amount per line', '11 476.44', and '531.91'. Below these columns is a large 'Result' area, which is currently empty. At the bottom of the interface, there is a footer that reads: 'Design & code: CMG Solutions — "Power BI fueled by AI"'. The entire interface is set against a dark blue background.

DEMO 4

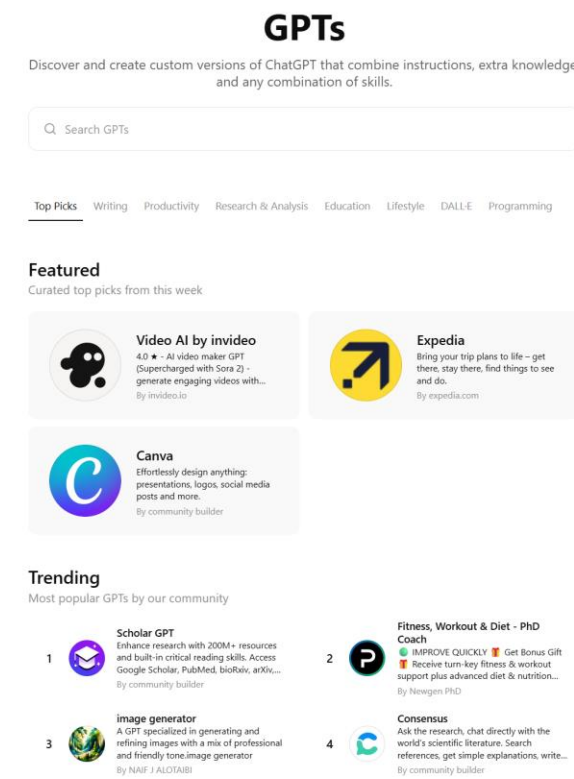
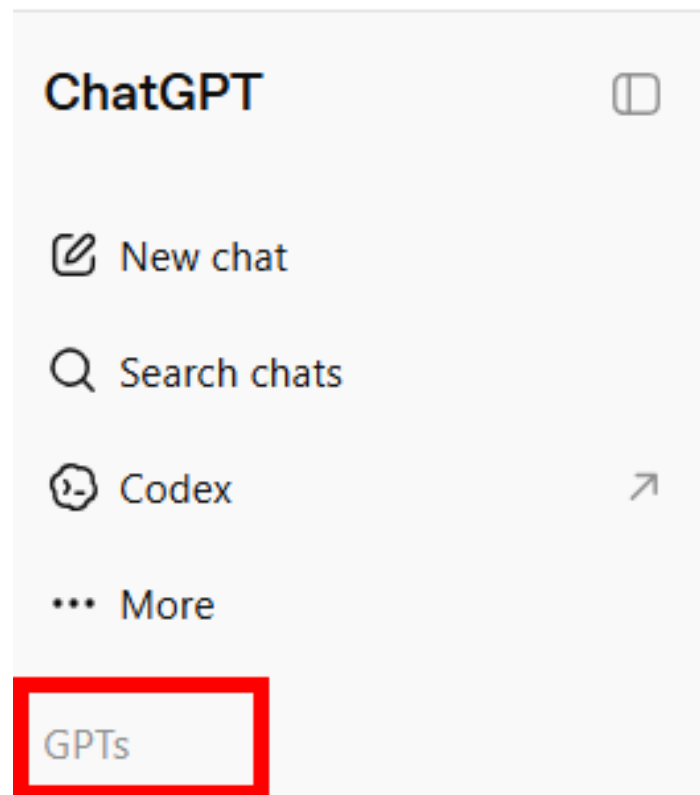
OCR & Barcode Generation with GPT

From any document to structured data and barcode, powered by a GPT



DEMO 4

OCR & Barcode Generation with GPT




QR Code Maker & Scanner Reader: Barcode

By accuratescribe.ai

QR Code Maker & Scanner Reader creates, scans, and manages QR codes or barcodes with ease. Batch-scan items, join Wi-Fi networks, share vCards, and organize unlimited code history and folders. Get instant, secure, and hassle-free code handling—no extra device needed.

DEMO 4

OCR & Barcode Generation with GPT

 sample-pdf-invoice.pdf
PDF

please create 3 bar codes from the attached pdf:
invoice date
invoice amount
invoice number

Thought for 14s >

Temporary Note Do Not Use

Copy Edit Download

Generating barcode image files from the uploaded invoice values.



DEMO 4

OCR & Barcode Generation with GPT

```
Invoice Barcodes - html Copy Edit Download Preview
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Invoice Barcodes</title>
<style>
body {
font-family: Arial, sans-serif;
background: #ff67fb;
color: #111827;
margin: 0;
padding: 32px;
}
.wrap {
max-width: 980px;
margin: 0 auto;
}
h1 {
margin: 0 0 8px;
font-size: 32px;
}
p.note {
margin: 0 0 24px;
color: #4b5563;
}
.card {
background: white;
border-radius: 20px;
padding: 24px;
margin-bottom: 20px;
box-shadow: 0 8px 24px rgba(0,0,0,0.08);
}
.label {
font-size: 14px;
text-transform: uppercase;
letter-spacing: 0.08em;
color: #6b7280;
margin-bottom: 8px;
}
.value {
font-size: 24px;
font-weight: 700;
margin-bottom: 16px;
}
svg {
width: 100%;
height: 140px;
display: block;
background: #fff;
}
.caption {
margin-top: 10px;
}
</style>
</html>
```

```
font-family: monospace;
font-size: 16px;
color: #111827;
}
.footer {
margin-top: 20px;
color: #6b7280;
font-size: 14px;
}
</style>
</head>
<body>
<div class="wrap">
<h1>Invoice Barcodes</h1>
<p class="note">Code 39 barcode set created from the uploaded invoice fields.</p>
<div class="card">
<div class="label">Invoice date</div>
<div class="value">2025-06-21</div>
<svg id="barcode-date" viewBox="0 0 1200 140" preserveAspectRatio="none"></svg>
<div class="caption">*2025-06-21*</div>
</div>
<div class="card">
<div class="label">Invoice amount</div>
<div class="value">1673.01</div>
<svg id="barcode-amount" viewBox="0 0 1200 140" preserveAspectRatio="none"></svg>
<div class="caption">*1673.01*</div>
</div>
<div class="card">
<div class="label">Invoice number</div>
<div class="value">90000001620</div>
<svg id="barcode-number" viewBox="0 0 1200 140" preserveAspectRatio="none"></svg>
<div class="caption">*90000001620*</div>
</div>
<div class="Footer">Barcode type: Code 39</div>
</div>
<script>
const CODE39 = {
'0': 'nnnwwnnn',
'1': 'wnnnnnnw',
'2': 'nnwwnnnw',
'3': 'wnwwnnnw',
'4': 'nnnwnnnw',
'5': 'wnnwnnnw',
'6': 'nnwwnnnw',
'7': 'nnnnnnnw',
'8': 'wnnnnnnw',
'9': 'nnwwnnnw',
'-': 'nnnnnnnw',
':': 'wnnnnnnw',
'.': 'nnwwnnnw',
'*': 'nnwwnnnw'
};
```

```
function drawCode39(svgId, value) {
const svg = document.getElementById(svgId);
const full = `${value}`;
const narrow = 8;
const wide = 20;
const height = 110;
let x = 20;

while (svg.firstChild) svg.removeChild(svg.firstChild);

const bg = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
bg.setAttribute('x', '0');
bg.setAttribute('y', '0');
bg.setAttribute('width', '1200');
bg.setAttribute('height', '140');
bg.setAttribute('fill', 'white');
svg.appendChild(bg);

for (let i = 0; i < full.length; i++) {
const ch = full[i];
const pattern = CODE39[ch];
if (!pattern) throw new Error('Unsupported Code 39 character: ${ch}');

for (let j = 0; j < pattern.length; j++) {
const isBar = j % 2 === 0;
const width = pattern[j] === 'w' ? wide : narrow;
if (isBar) {
const rect = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
rect.setAttribute('x', String(x));
rect.setAttribute('y', '10');
rect.setAttribute('width', String(width));
rect.setAttribute('height', String(height));
rect.setAttribute('fill', '#000');
svg.appendChild(rect);
}
x += width;
}

if (i < full.length - 1) {
x += narrow;
}

const text = document.createElementNS('http://www.w3.org/2000/svg', 'text');
text.setAttribute('x', '50%');
text.setAttribute('y', '136');
text.setAttribute('text-anchor', 'middle');
text.setAttribute('font-size', '18');
text.setAttribute('font-family', 'monospace');
text.textContent = full;
svg.appendChild(text);
}

drawCode39('barcode-date', '2025-06-21');
drawCode39('barcode-amount', '1673.01');
drawCode39('barcode-number', '90000001620');
</script>
```

```
</body>
</html>
```

I created a previewable HTML file with 3 Code 39 barcodes for:

- Invoice date: 2025-06-21
- Invoice amount: 1673.01
- Invoice number: 90000001620

Those values were taken from the uploaded invoice PDF.

The barcode file is open in the canvas as Invoice Barcodes.

Discover more at <https://apps.apple.com/app/id6451921394>.



DEMO 4

OCR & Barcode Generation with GPT

Invoice Barcodes

Code 39 barcode set created from the uploaded invoice fields.

INVOICE DATE

2025-06-21



2025-06-21

INVOICE AMOUNT

1673.01



1673.01

INVOICE NUMBER

90000001620



90000001620

Canada Post logo and header information including invoice date (2025-06-21), customer number (90000001620), and contact details.

Summary of charges

	Subtotal	GST	HST (page 2)	PST/QST	Total
Parcels	97.98				97.98
Commercial/Smartmail Marketing	216.90				216.90
Specialized services	1,240.00				1,240.00
Shipments	168.13				168.13
Adjustments	-50.00				-50.00
Total	1,673.01				\$ 1,673.01

Canada Post registration nos.: HST/GST 119321495; QST 100699299

Return this slip with your payment

	Customer number	Amount due
	1234567	\$ 1,673.01
	Invoice number 90000001620	Due date (Y-M-D) 2025-07-06

15 days from date of invoice. Late payment fees at 18% per annum will be charged on any invoice paid after due date.

Amount paid \$

0000007276697090000001620000167301 001

DEMO 5

ChatGPT in Action

Testing OCR & Barcode Creation



From **unstructured** to **actionable**.

Some Reports I Created with Power BI



Power BI

- Ajustement
- Avocat ext.
- Carte crédit
- Dépenses crédit
- Intermédiaire pmt

USE AI TO CREATE YOUR AP TEAM'S SUCCESS STORY

Create a one-page report that highlights your impact, showcases your wins, and gets your team noticed.

WHY IT MATTERS



GET RECOGNIZED
Showcase your team's contributions and the value you deliver.



DRIVE IMPACT
Quantify results that support your goals and the business.



BUILD VISIBILITY
Make your team's work visible across the organization.



AI DOES THE HEAVY LIFTING
Turn your data into a clear, professional one-pager—fast.

- ✓ Analyze your data
- ✓ Highlight key achievements
- ✓ Generate a compelling report

ACCOUNTS PAYABLE DEPARTMENT

DELIVERING VALUE. DRIVING RESULTS.

2025

P2P CYCLE TIME

-18%

faster cycle time
vs. 2024



EARLY PAYMENT DISCOUNTS CAPTURED

\$1.2M

in savings



ON-TIME PAYMENT RATE

99.2%

suppliers paid on time



STRAIGHT-THROUGH PROCESSING

78%

touchless invoices



KEY ACHIEVEMENTS

- ✓ Improved cash flow and liquidity
- ✓ Strengthened supplier relationships
- ✓ Increased efficiency through automation
- ✓ Reduced exceptions and manual work
- ✓ Supported compliance and control

OUR IMPACT



We keep the business moving forward.
We deliver value every day.



ONE TEAM. REAL RESULTS.
THANK YOU FOR SUPPORTING AP!

AI in action:

What I Actually Do With it

MOVE FASTER

- 1 Draft emails and translate content
- 2 Learn new tools (ex: Power BI, SAP, Excel)
- 3 First line of support for IT and SAP troubleshooting

THINK SMARTER

- 4 Generate and optimize DAX formulas in Power BI
- 5 Summarize and analyze large datasets or documents
- 6 Brainstorm ideas and validate approaches

BUILD & CREATE

- 7 Build simple tools (HTML, JavaScript) to solve real problems
- 8 Create dashboards and visuals for decision-making
- 9 Support presentation creation (content, structure, visuals)

 **Not theory. Real daily use.** | **Used every week. Not once a year.**

Lessons Learned & Key Takeaways



1 Start small, test fast

Don't aim for perfect.
Test simple ideas quickly,
learn, adjust, and move on.



2 Adoption beats perfection

If people don't use it, it has no value.
Build solutions inside everyday
tools (PowerBI, email, ERP).



3 Solve real problems first

Focus on concrete,
repetitive pain points.
Not everything needs AI.



4 Keep it simple

Complex solutions
create new problems.



5 Time is the real constraint

AI doesn't create value on its own.
Freeing time does.



Focus on people, problems and time — **AI is the enabler, not the goal.**

Final Thoughts



Keep human in the loop!

Judgment, accountability, and context remain human.



AI does not replace people.

People using AI will replace those who don't.

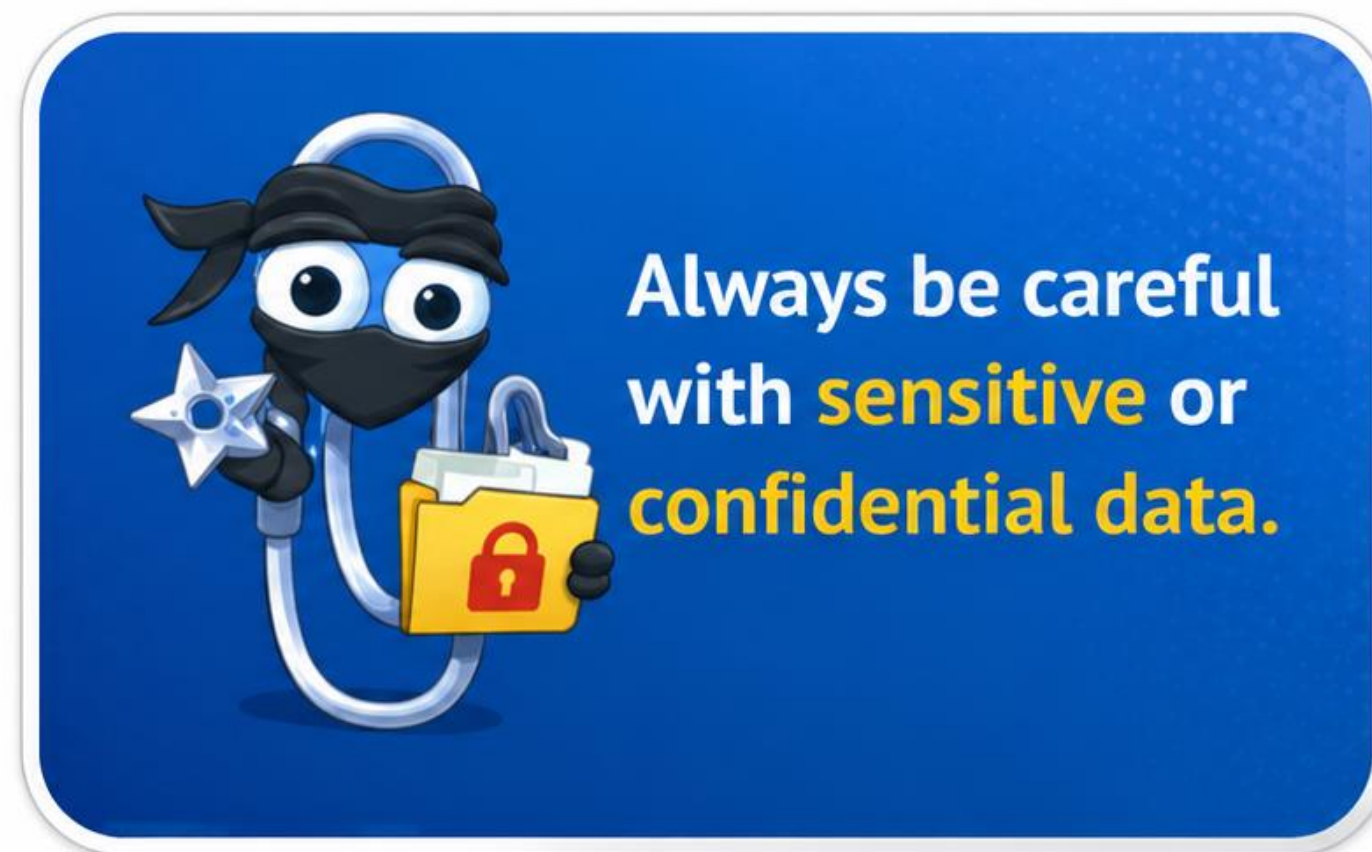


Professionals who use AI well gain speed, focus, and insight.

They do this by **removing repetitive work** and freeing time for **higher-value tasks**.



Use AI responsibly. Focus on **people, problems** and **value** — not just **technology**.



Please tell us what you think!

- Please scan this QR code using your mobile to access a short feedback survey →
- Also accessible via the mobile app



Tues 3:00 | No Budget, No Problem: Building Custom AI Tools To Improve AP Efficiency

IOFM Spring
CONFERENCE & EXPO



QUESTIONS?

REMINDER!

If you checked in for NASBA CPE credit, check out at iofm.cnf.io