PCBancard Proposal Generator — Visual Upgrade (Hybrid Approach)

OVERVIEW

This upgrade improves the visual quality of generated proposals WITHOUT breaking existing functionality. It adds a new rendering pipeline that produces beautiful, professionally-designed PDFs using styled HTML + Puppeteer.

What changes:

- Claude generates structured JSON (not just Markdown)
- A pre-designed HTML template renders the data beautifully
- Gemini generates custom images for each proposal
- Puppeteer converts the final HTML to PDF

What stays the same:

- All existing form fields and data collection
- Spreadsheet parsing logic
- Business research functionality
- The data pipeline (just adding a better renderer at the end)

STEP 1: Install Required Dependencies

```
npm install puppeteer cheerio
```

If Puppeteer has issues in Replit, use `puppeteer-core` with a hosted Chrome instance, or fall back to `html-pdf-node` .

STEP 2: Create the HTML Template

Create a new file: `/templates/proposal-template.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{business_name}} - Payment Processing Proposal</title>
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: 'Inter', -apple-system, BlinkMacSystemFont, sans-serif;
      color: #1a1a2e;
      line-height: 1.6;
      background: #ffffff;
    }

    .page {
      width: 8.5in;
      min-height: 11in;
      padding: 0.5in;
      page-break-after: always;
    }

    .page:last-child {
      page-break-after: avoid;
    }

    /* ========== COVER PAGE ========== */
    .cover-page {
      display: flex;
      flex-direction: column;
      justify-content: space-between;
      min-height: 10in;
      background: linear-gradient(180deg, #ffffff 0%, #f8fafc 100%);
    }

    .cover-header {
      display: flex;
      justify-content: space-between;
      align-items: flex-start;
      padding-bottom: 20px;
      border-bottom: 1px solid #e2e8f0;
    }

    .pcbancard-logo {
      height: 40px;
    }

    .cover-hero {
      flex: 1;
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      text-align: center;
      padding: 60px 40px;
    }

    .hero-image {
      width: 100%;
      max-width: 600px;
      height: 200px;
      object-fit: cover;
      border-radius: 16px;
      margin-bottom: 40px;
      box-shadow: 0 20px 40px rgba(30, 58, 95, 0.15);
    }

    .merchant-logo {
      max-height: 80px;
      max-width: 200px;
      margin-bottom: 24px;
    }

    .cover-title {
      font-size: 36px;
      font-weight: 800;
      color: #1E3A5F;
      margin-bottom: 8px;
      letter-spacing: -0.5px;
    }

    .cover-subtitle {
      font-size: 20px;
      color: #64748b;
      font-weight: 500;
```

```
}

.cover-footer {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 40px;
  padding-top: 40px;
  border-top: 2px solid #1E3A5F;
}

.cover-section h4 {
  font-size: 11px;
  text-transform: uppercase;
  letter-spacing: 1px;
  color: #94a3b8;
  margin-bottom: 8px;
}

.cover-section p {
  font-size: 14px;
  color: #334155;
  margin: 2px 0;
}

.cover-section .name {
  font-size: 16px;
  font-weight: 600;
  color: #1E3A5F;
}

/* ========== SECTION HEADERS ========== */
.section-header {
  display: flex;
  align-items: center;
  margin-bottom: 24px;
  padding-bottom: 12px;
  border-bottom: 3px solid #1E3A5F;
}

.section-number {
  width: 36px;
  height: 36px;
  background: linear-gradient(135deg, #1E3A5F 0%, #2d4a6f 100%);
  color: white;
  border-radius: 50%;
  display: flex;
  align-items: center;
```

```css
  justify-content: center;
  font-weight: 700;
  font-size: 16px;
  margin-right: 16px;
}

.section-title {
  font-size: 24px;
  font-weight: 700;
  color: #1E3A5F;
}

/* ========== EXECUTIVE SUMMARY ========== */
.executive-summary {
  padding: 32px;
  background: linear-gradient(135deg, #f0f9ff 0%, #e0f2fe 100%);
  border-radius: 16px;
  margin-bottom: 32px;
  border-left: 4px solid #1E3A5F;
}

.executive-summary h3 {
  font-size: 18px;
  color: #1E3A5F;
  margin-bottom: 12px;
}

.executive-summary p {
  font-size: 15px;
  color: #334155;
}

.highlight-box {
  background: white;
  padding: 20px;
  border-radius: 12px;
  margin-top: 20px;
  box-shadow: 0 4px 12px rgba(0,0,0,0.05);
}

.highlight-box strong {
  color: #1E3A5F;
}

/* ========== SAVINGS CARDS ========== */
.savings-grid {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 24px;
  margin: 32px 0;
}

.savings-card {
  padding: 32px;
  border-radius: 16px;
  position: relative;
  overflow: hidden;
}

.savings-card::before {
  content: '';
  position: absolute;
  top: -50px;
  right: -50px;
  width: 150px;
  height: 150px;
  border-radius: 50%;
  opacity: 0.15;
}

.savings-card.dual-pricing {
  background: linear-gradient(135deg, #ecfdf5 0%, #d1fae5 100%);
  border: 1px solid #a7f3d0;
}

.savings-card.dual-pricing::before {
  background: #10b981;
}

.savings-card.interchange-plus {
  background: linear-gradient(135deg, #eff6ff 0%, #dbeafe 100%);
  border: 1px solid #93c5fd;
}

.savings-card.interchange-plus::before {
  background: #3b82f6;
}

.savings-card .card-label {
  font-size: 14px;
  font-weight: 600;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  margin-bottom: 8px;
```

```
}

.savings-card.dual-pricing .card-label {
  color: #059669;
}

.savings-card.interchange-plus .card-label {
  color: #2563eb;
}

.savings-card .savings-amount {
  font-size: 42px;
  font-weight: 800;
  line-height: 1;
  margin-bottom: 4px;
}

.savings-card.dual-pricing .savings-amount {
  color: #047857;
}

.savings-card.interchange-plus .savings-amount {
  color: #1d4ed8;
}

.savings-card .savings-period {
  font-size: 16px;
  opacity: 0.8;
}

.savings-card .monthly-detail {
  margin-top: 20px;
  padding-top: 16px;
  border-top: 1px solid rgba(0,0,0,0.1);
  font-size: 14px;
  color: #64748b;
}

.savings-card .monthly-detail strong {
  color: #334155;
}

/* ========== DATA TABLES ========== */
.data-table {
  width: 100%;
  border-collapse: collapse;
  margin: 24px 0;
```

```css
  font-size: 14px;
  box-shadow: 0 4px 12px rgba(0,0,0,0.08);
  border-radius: 12px;
  overflow: hidden;
}

.data-table thead {
  background: linear-gradient(135deg, #1E3A5F 0%, #2d4a6f 100%);
  color: white;
}

.data-table th {
  padding: 16px 20px;
  text-align: left;
  font-weight: 600;
  font-size: 13px;
  text-transform: uppercase;
  letter-spacing: 0.5px;
}

.data-table td {
  padding: 16px 20px;
  border-bottom: 1px solid #e2e8f0;
}

.data-table tbody tr:nth-child(even) {
  background: #f8fafc;
}

.data-table tbody tr:hover {
  background: #f1f5f9;
}

.data-table .highlight-row {
  background: #ecfdf5 !important;
  font-weight: 600;
}

.data-table .highlight-row td {
  color: #047857;
}

.data-table .number {
  font-family: 'SF Mono', 'Fira Code', monospace;
  font-weight: 600;
}
```

```
.data-table .positive {
  color: #059669;
}

.data-table .negative {
  color: #dc2626;
}

/* ========== COMPARISON TABLE ========== */
.comparison-table {
  width: 100%;
  border-collapse: separate;
  border-spacing: 0;
  margin: 32px 0;
  border-radius: 16px;
  overflow: hidden;
  box-shadow: 0 8px 24px rgba(0,0,0,0.1);
}

.comparison-table thead {
  background: #1E3A5F;
  color: white;
}

.comparison-table th {
  padding: 20px 24px;
  font-weight: 600;
  font-size: 14px;
}

.comparison-table th:first-child {
  text-align: left;
}

.comparison-table td {
  padding: 20px 24px;
  text-align: center;
  border-bottom: 1px solid #e2e8f0;
  font-size: 15px;
}

.comparison-table td:first-child {
  text-align: left;
  font-weight: 500;
  color: #475569;
}
```

```css
.comparison-table tbody tr:last-child td {
  border-bottom: none;
}

.comparison-table .current-col {
  background: #fef2f2;
  color: #991b1b;
}

.comparison-table .dp-col {
  background: #ecfdf5;
  color: #047857;
  font-weight: 700;
}

.comparison-table .icp-col {
  background: #eff6ff;
  color: #1d4ed8;
  font-weight: 600;
}

.comparison-table .winner {
  position: relative;
}

.comparison-table .winner::after {
  content: '★ BEST';
  position: absolute;
  top: 4px;
  right: 4px;
  font-size: 9px;
  background: #fbbf24;
  color: #78350f;
  padding: 2px 6px;
  border-radius: 4px;
  font-weight: 700;
}

/* ========== RECOMMENDATION BOX ========== */
.recommendation-box {
  background: linear-gradient(135deg, #fef3c7 0%, #fde68a 100%);
  border: 2px solid #f59e0b;
  border-radius: 16px;
  padding: 32px;
  margin: 32px 0;
}
```

```
.recommendation-box h3 {
  color: #92400e;
  font-size: 20px;
  margin-bottom: 16px;
  display: flex;
  align-items: center;
  gap: 12px;
}

.recommendation-box h3::before {
  content: '💡';
  font-size: 24px;
}

.recommendation-box p {
  color: #78350f;
  font-size: 15px;
}

.recommendation-reasons {
  margin-top: 20px;
  display: grid;
  gap: 12px;
}

.reason-item {
  display: flex;
  align-items: flex-start;
  gap: 12px;
  background: white;
  padding: 16px;
  border-radius: 8px;
}

.reason-number {
  width: 24px;
  height: 24px;
  background: #f59e0b;
  color: white;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 12px;
  font-weight: 700;
  flex-shrink: 0;
}
```

```
.reason-text {
  font-size: 14px;
  color: #78350f;
}

/* ========== EQUIPMENT SECTION ========== */
.equipment-card {
  display: grid;
  grid-template-columns: 200px 1fr;
  gap: 32px;
  background: #f8fafc;
  border-radius: 16px;
  padding: 32px;
  margin: 24px 0;
  border: 1px solid #e2e8f0;
}

.equipment-image {
  width: 100%;
  height: auto;
  border-radius: 12px;
  box-shadow: 0 8px 24px rgba(0,0,0,0.1);
}

.equipment-details h3 {
  font-size: 24px;
  color: #1E3A5F;
  margin-bottom: 8px;
}

.equipment-details .tagline {
  color: #64748b;
  font-size: 15px;
  margin-bottom: 20px;
}

.feature-list {
  list-style: none;
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 12px;
}

.feature-list li {
  display: flex;
  align-items: center;
  gap: 8px;
  font-size: 14px;
  color: #475569;
}

.feature-list li::before {
  content: '✓';
  color: #10b981;
  font-weight: bold;
}

/* ========== TIMELINE ========== */
.timeline {
  margin: 32px 0;
}

.timeline-item {
  display: grid;
  grid-template-columns: 80px 40px 1fr;
  gap: 16px;
  margin-bottom: 24px;
}

.timeline-day {
  text-align: right;
  font-weight: 700;
  color: #1E3A5F;
  font-size: 14px;
}

.timeline-dot {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.timeline-dot::before {
  content: '';
  width: 16px;
  height: 16px;
  background: linear-gradient(135deg, #1E3A5F 0%, #3b82f6 100%);
  border-radius: 50%;
  border: 3px solid #e0f2fe;
}

.timeline-dot::after {
  content: '';
  width: 2px;
  flex: 1;
  background: #cbd5e1;
  margin-top: 8px;
}

.timeline-item:last-child .timeline-dot::after {
  display: none;
}

.timeline-content {
  background: white;
  padding: 16px 20px;
  border-radius: 12px;
  border: 1px solid #e2e8f0;
  box-shadow: 0 2px 8px rgba(0,0,0,0.04);
}

.timeline-content h4 {
  color: #1E3A5F;
  font-size: 15px;
  margin-bottom: 4px;
}

.timeline-content p {
  color: #64748b;
  font-size: 13px;
}

/* ========== CTA SECTION ========== */
.cta-section {
  background: linear-gradient(135deg, #1E3A5F 0%, #0f172a 100%);
  color: white;
  padding: 48px;
  border-radius: 20px;
  text-align: center;
  margin: 40px 0;
}

.cta-section h2 {
  font-size: 28px;
  margin-bottom: 12px;
}

.cta-section p {
  font-size: 16px;
  opacity: 0.9;
  margin-bottom: 32px;
}

.cta-buttons {
  display: flex;
  justify-content: center;
  gap: 16px;
  flex-wrap: wrap;
}

.cta-button {
  padding: 16px 32px;
  border-radius: 8px;
  font-size: 16px;
  font-weight: 600;
  text-decoration: none;
  transition: transform 0.2s;
}

.cta-button.primary {
  background: #2ECC71;
  color: white;
}

.cta-button.secondary {
  background: rgba(255,255,255,0.1);
  color: white;
  border: 1px solid rgba(255,255,255,0.3);
}

.agent-contact {
  margin-top: 32px;
  padding-top: 24px;
  border-top: 1px solid rgba(255,255,255,0.2);
}

.agent-contact p {
  margin: 4px 0;
  font-size: 14px;
}

.agent-contact .agent-name {
  font-size: 18px;
  font-weight: 600;
}

/* ========== DISCLOSURES ========== */
```

```
    .disclosures {
      background: #f8fafc;
      padding: 24px;
      border-radius: 12px;
      margin-top: 40px;
    }

    .disclosures h4 {
      font-size: 14px;
      color: #64748b;
      margin-bottom: 12px;
      text-transform: uppercase;
      letter-spacing: 0.5px;
    }

    .disclosures ul {
      list-style: none;
    }

    .disclosures li {
      font-size: 12px;
      color: #94a3b8;
      padding: 6px 0;
      border-bottom: 1px solid #e2e8f0;
    }

    .disclosures li:last-child {
      border-bottom: none;
    }

    /* ========== PRINT STYLES ========== */
    @media print {
      .page {
        page-break-after: always;
      }

      body {
        -webkit-print-color-adjust: exact !important;
        print-color-adjust: exact !important;
      }
    }
  </style>
</head>
<body>

  <!-- ==================== PAGE 1: COVER ==================== -->
  <div class="page cover-page">
    <div class="cover-header">
      <img src="{{pcbancard_logo_url}}" alt="PCBancard" class="pcbancard-logo">
      <span style="font-size: 12px; color: #94a3b8;">Confidential Proposal</span>
    </div>

    <div class="cover-hero">
      {{#if hero_image_url}}
      <img src="{{hero_image_url}}" alt="Hero" class="hero-image">
      {{/if}}

      {{#if merchant_logo_url}}
      <img src="{{merchant_logo_url}}" alt="{{business_name}}" class="merchant-lo
      {{/if}}

      <h1 class="cover-title">Custom Payment Processing Proposal</h1>
      <p class="cover-subtitle">Prepared exclusively for {{business_name}}</p>
    </div>

    <div class="cover-footer">
      <div class="cover-section">
        <h4>Prepared For</h4>
        <p class="name">{{business_name}}</p>
        {{#if owner_name}}<p>{{owner_name}}</p>{{/if}}
        {{#if business_address}}<p>{{business_address}}</p>{{/if}}
        {{#if business_phone}}<p>{{business_phone}}</p>{{/if}}
        {{#if business_email}}<p>{{business_email}}</p>{{/if}}
      </div>
      <div class="cover-section" style="text-align: right;">
        <h4>Prepared By</h4>
        <p class="name">{{agent_first_name}} {{agent_last_name}}</p>
        <p>{{agent_title}}</p>
        <p>{{agent_phone}}</p>
        <p>{{agent_email}}</p>
        <p style="margin-top: 12px; color: #94a3b8;">{{proposal_date}}</p>
      </div>
    </div>
  </div>

  <!-- ==================== PAGE 2: EXECUTIVE SUMMARY ==================== -->
  <div class="page">
    <div class="section-header">
      <div class="section-number">1</div>
      <h2 class="section-title">Executive Summary</h2>
    </div>

    <div class="executive-summary">
      <h3>About {{business_name}}</h3>
      <p>{{business_description}}</p>

      <div class="highlight-box">
        <strong>The Opportunity:</strong> {{opportunity_statement}}
      </div>
    </div>

    <div class="section-header" style="margin-top: 40px;">
      <div class="section-number">2</div>
      <h2 class="section-title">Your Potential Savings</h2>
    </div>

    <div class="savings-grid">
      <div class="savings-card dual-pricing">
        <div class="card-label">Dual Pricing</div>
        <div class="savings-amount">${{dp_annual_savings}}</div>
        <div class="savings-period">potential annual savings</div>
        <div class="monthly-detail">
          Monthly cost: <strong>${{dp_monthly_cost}}</strong><br>
          Monthly savings: <strong>${{dp_monthly_savings}}</strong>
        </div>
      </div>

      <div class="savings-card interchange-plus">
        <div class="card-label">Interchange Plus</div>
        <div class="savings-amount">${{icp_annual_savings}}</div>
        <div class="savings-period">potential annual savings</div>
        <div class="monthly-detail">
          Monthly cost: <strong>${{icp_monthly_cost}}</strong><br>
          Monthly savings: <strong>${{icp_monthly_savings}}</strong>
        </div>
      </div>
    </div>

    <p style="text-align: center; color: #64748b; font-size: 14px; margin-top: 24
      Based on your current monthly volume of <strong>${{current_monthly_volume}}
      and processing fees of <strong>${{current_monthly_fees}}</strong>
    </p>
  </div>

  <!-- ==================== PAGE 3: COMPARISON ==================== -->
  <div class="page">
    <div class="section-header">
      <div class="section-number">3</div>
      <h2 class="section-title">Current Processing Snapshot</h2>
    </div>
```

```
<table class="data-table">
  <thead>
    <tr>
      <th>Metric</th>
      <th>Current Value</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Monthly Processing Volume</td>
      <td class="number">${{current_monthly_volume}}</td>
    </tr>
    <tr>
      <td>Monthly Transactions</td>
      <td class="number">{{current_transactions}}</td>
    </tr>
    <tr>
      <td>Average Ticket Size</td>
      <td class="number">${{current_avg_ticket}}</td>
    </tr>
    <tr>
      <td>Current Monthly Fees</td>
      <td class="number negative">${{current_monthly_fees}}</td>
    </tr>
    <tr>
      <td>Effective Rate</td>
      <td class="number">{{current_effective_rate}}%</td>
    </tr>
    <tr>
      <td>Annual Processing Cost</td>
      <td class="number negative">${{current_annual_fees}}</td>
    </tr>
  </tbody>
</table>

<div class="section-header" style="margin-top: 48px;">
  <div class="section-number">4</div>
  <h2 class="section-title">Side-by-Side Comparison</h2>
</div>

<table class="comparison-table">
  <thead>
    <tr>
      <th></th>
      <th>Current</th>
      <th>Dual Pricing</th>
      <th>Interchange+</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Monthly Cost</td>
        <td class="current-col">${{current_monthly_fees}}</td>
        <td class="dp-col {{#if dp_is_winner}}winner{{/if}}">${{dp_monthly_cost
        <td class="icp-col">${{icp_monthly_cost}}</td>
      </tr>
      <tr>
        <td>Annual Cost</td>
        <td class="current-col">${{current_annual_fees}}</td>
        <td class="dp-col">${{dp_annual_cost}}</td>
        <td class="icp-col">${{icp_annual_cost}}</td>
      </tr>
      <tr>
        <td>Monthly Savings</td>
        <td class="current-col">—</td>
        <td class="dp-col">${{dp_monthly_savings}}</td>
        <td class="icp-col">${{icp_monthly_savings}}</td>
      </tr>
      <tr>
        <td>Annual Savings</td>
        <td class="current-col">—</td>
        <td class="dp-col">${{dp_annual_savings}}</td>
        <td class="icp-col">${{icp_annual_savings}}</td>
      </tr>
    </tbody>
  </table>

  <div class="recommendation-box">
    <h3>Our Recommendation: {{recommended_option}}</h3>
    <p>{{recommendation_summary}}</p>

    <div class="recommendation-reasons">
      {{#each recommendation_reasons}}
      <div class="reason-item">
        <div class="reason-number">{{@index_plus_one}}</div>
        <div class="reason-text">{{this}}</div>
      </div>
      {{/each}}
    </div>
  </div>
</div>

<!-- ==================== PAGE 4: EQUIPMENT & NEXT STEPS ==================== -
<div class="page">
```

```
<div class="section-header">
  <div class="section-number">5</div>
  <h2 class="section-title">Recommended Equipment</h2>
</div>

<div class="equipment-card">
  {{#if equipment_image_url}}
  <img src="{{equipment_image_url}}" alt="{{equipment_name}}" class="equipmen
  {{else}}
  <div style="background: #e2e8f0; border-radius: 12px; display: flex; align-
    Equipment Image
  </div>
  {{/if}}

  <div class="equipment-details">
    <h3>{{equipment_name}}</h3>
    <p class="tagline">{{equipment_tagline}}</p>

    <ul class="feature-list">
      {{#each equipment_features}}
      <li>{{this}}</li>
      {{/each}}
    </ul>
  </div>
</div>

<div class="section-header" style="margin-top: 48px;">
  <div class="section-number">6</div>
  <h2 class="section-title">Implementation Timeline</h2>
</div>

<div class="timeline">
  {{#each timeline_steps}}
  <div class="timeline-item">
    <div class="timeline-day">{{this.day}}</div>
    <div class="timeline-dot"></div>
    <div class="timeline-content">
      <h4>{{this.title}}</h4>
      <p>{{this.description}}</p>
    </div>
  </div>
  {{/each}}
</div>

<div class="cta-section">
  <h2>Ready to Start Saving?</h2>
  <p>Let's schedule a quick call to finalize your setup and get you saving.</
```

```
      <div class="cta-buttons">
        <a href="#" class="cta-button primary">Schedule a Call</a>
        <a href="#" class="cta-button secondary">Reply: "YES — send next steps"</
      </div>

      <div class="agent-contact">
        <p class="agent-name">{{agent_first_name}} {{agent_last_name}}</p>
        <p>{{agent_title}}</p>
        <p>{{agent_phone}} | {{agent_email}}</p>
      </div>
    </div>

    <div class="disclosures">
      <h4>Compliance &amp; Disclosures</h4>
      <ul>
        {{#each disclosures}}
        <li>{{this}}</li>
        {{/each}}
      </ul>
    </div>
  </div>

</body>
</html>
```

STEP 3: Create the Template Renderer

Create a new file: `/renderers/html-renderer.js`

```
const fs = require('fs');
const path = require('path');
const Handlebars = require('handlebars');
const puppeteer = require('puppeteer');

// Register Handlebars helpers
Handlebars.registerHelper('index_plus_one', function(index) {
  return index + 1;
});

class HtmlRenderer {
  constructor() {
    this.templatePath = path.join(__dirname, '../templates/proposal-template.html
    this.template = null;
```

```
}

loadTemplate() {
  if (!this.template) {
    const templateSource = fs.readFileSync(this.templatePath, 'utf8');
    this.template = Handlebars.compile(templateSource);
  }
  return this.template;
}

/**
 * Render proposal data to HTML
 */
renderHtml(proposalData) {
  const template = this.loadTemplate();

  // Ensure all required fields have values or fallbacks
  const data = this.prepareData(proposalData);

  return template(data);
}

/**
 * Prepare and validate data for template
 */
prepareData(raw) {
  return {
    // Branding
    pcbancard_logo_url: raw.pcbancard_logo_url || '/assets/pcbancard-logo.png',
    hero_image_url: raw.hero_image_url || null,
    merchant_logo_url: raw.merchant_logo_url || null,

    // Merchant info
    business_name: raw.business_name || 'Valued Merchant',
    owner_name: raw.owner_name || null,
    business_address: raw.business_address || null,
    business_phone: raw.business_phone || null,
    business_email: raw.business_email || null,
    business_description: raw.business_description || `${raw.business_name} is
    opportunity_statement: raw.opportunity_statement || 'Reduce payment process

    // Agent info
    agent_first_name: raw.agent_first_name || '',
    agent_last_name: raw.agent_last_name || '',
    agent_title: raw.agent_title || 'Account Executive',
    agent_phone: raw.agent_phone || '',
    agent_email: raw.agent_email || '',
// Date
proposal_date: raw.proposal_date || new Date().toLocaleDateString('en-US',
  year: 'numeric', month: 'long', day: 'numeric'
}),

// Current state (formatted)
current_monthly_volume: this.formatCurrency(raw.current_monthly_volume),
current_transactions: this.formatNumber(raw.current_transactions),
current_avg_ticket: this.formatCurrency(raw.current_avg_ticket),
current_monthly_fees: this.formatCurrency(raw.current_monthly_fees),
current_effective_rate: this.formatPercent(raw.current_effective_rate),
current_annual_fees: this.formatCurrency(raw.current_annual_fees),

// Dual Pricing (formatted)
dp_monthly_cost: this.formatCurrency(raw.dp_monthly_cost),
dp_annual_cost: this.formatCurrency(raw.dp_annual_cost || (raw.dp_monthly_c
dp_monthly_savings: this.formatCurrency(raw.dp_monthly_savings),
dp_annual_savings: this.formatCurrency(raw.dp_annual_savings),
dp_is_winner: raw.dp_monthly_savings > raw.icp_monthly_savings,

// Interchange Plus (formatted)
icp_monthly_cost: this.formatCurrency(raw.icp_monthly_cost),
icp_annual_cost: this.formatCurrency(raw.icp_annual_cost || (raw.icp_monthl
icp_monthly_savings: this.formatCurrency(raw.icp_monthly_savings),
icp_annual_savings: this.formatCurrency(raw.icp_annual_savings),

// Recommendation
recommended_option: raw.recommended_option || 'Dual Pricing',
recommendation_summary: raw.recommendation_summary || 'Based on your busine
recommendation_reasons: raw.recommendation_reasons || [
  'Maximum savings potential with minimal operational changes',
  'Customers appreciate transparency and choice in payment options',
  'Fully automated system requires no manual price adjustments'
],

// Equipment
equipment_name: raw.equipment_name || 'Recommended Terminal',
equipment_tagline: raw.equipment_tagline || 'Selected for your business nee
equipment_image_url: raw.equipment_image_url || null,
equipment_features: raw.equipment_features || [
  'EMV chip card support',
  'NFC/contactless payments',
  'Built-in receipt printer',
  'WiFi + Ethernet connectivity'
],
    // Timeline
    timeline_steps: raw.timeline_steps || [
      { day: 'Day 1', title: 'Consultation', description: 'Finalize program sel
      { day: 'Day 2-3', title: 'Setup', description: 'Equipment configuration a
      { day: 'Day 4', title: 'Installation', description: 'On-site installation
      { day: 'Day 5', title: 'Go Live', description: 'Begin processing with new
      { day: 'Day 30', title: 'Review', description: 'First month check-in and
    ],

    // Disclosures
    disclosures: raw.disclosures || [
      'All savings estimates based on provided processing data and stated assum
      'Actual results may vary based on card mix, transaction patterns, and cus
      'Dual pricing requires compliant signage and customer notification.',
      'Rates and fees subject to underwriting approval.',
      'This proposal is valid for 30 days from the preparation date.'
    ]
  };
}

/**
 * Format currency safely
 */
formatCurrency(value) {
  if (value === null || value === undefined || isNaN(value)) {
    return '—';
  }
  return new Intl.NumberFormat('en-US', {
    minimumFractionDigits: 2,
    maximumFractionDigits: 2
  }).format(value);
}

/**
 * Format number safely
 */
formatNumber(value) {
  if (value === null || value === undefined || isNaN(value)) {
    return '—';
  }
  return new Intl.NumberFormat('en-US').format(value);
}

/**
 * Format percent safely
 */
formatPercent(value) {
  if (value === null || value === undefined || isNaN(value)) {
    return '—';
  }
  return value.toFixed(2);
}

/**
 * Convert HTML to PDF using Puppeteer
 */
async htmlToPdf(htmlContent, outputPath) {
  const browser = await puppeteer.launch({
    headless: 'new',
    args: ['--no-sandbox', '--disable-setuid-sandbox']
  });

  try {
    const page = await browser.newPage();

    // Set content and wait for images to load
    await page.setContent(htmlContent, {
      waitUntil: ['load', 'networkidle0']
    });

    // Generate PDF
    await page.pdf({
      path: outputPath,
      format: 'Letter',
      printBackground: true,
      margin: {
        top: '0',
        right: '0',
        bottom: '0',
        left: '0'
      }
    });

    return outputPath;

  } finally {
    await browser.close();
  }
}

/**
 * Full render pipeline: Data → HTML → PDF
 */
async generateProposal(proposalData, outputPath) {
    // Step 1: Render HTML
    const html = this.renderHtml(proposalData);

    // Step 2: Convert to PDF
    await this.htmlToPdf(html, outputPath);

    return {
      success: true,
      path: outputPath,
      format: 'pdf'
    };
  }
}

module.exports = { HtmlRenderer };
```

STEP 4: Create the Image Generation Module

Create a new file: `/services/image-generator.js`

```
/**
 * Generate proposal images using Gemini (Nano Banana Pro)
 */
class ImageGenerator {
  constructor(apiKey) {
    this.apiKey = apiKey || process.env.GEMINI_API_KEY;
    this.baseUrl = 'https://generativelanguage.googleapis.com/v1beta';
  }

  /**
   * Generate a single image
   */
  async generateImage(prompt, style = 'professional') {
    // If no API key, return null (graceful degradation)
    if (!this.apiKey) {
      console.log('Gemini API key not configured, skipping image generation');
      return null;
    }

    try {
      const response = await fetch(`${this.baseUrl}/models/imagen-3.0:generateIma
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        'x-goog-api-key': this.apiKey
      },
      body: JSON.stringify({
        prompt: `${prompt}. Style: ${style}, high quality, professional, suitab
        numberOfImages: 1,
        aspectRatio: '16:9'
      })
    });

    if (!response.ok) {
      console.error('Image generation failed:', response.status);
      return null;
    }

    const data = await response.json();

    // Return base64 image data or URL depending on API response
    if (data.generatedImages && data.generatedImages[0]) {
      return data.generatedImages[0].image?.imageBytes
        ? `data:image/png;base64,${data.generatedImages[0].image.imageBytes}`
        : data.generatedImages[0].image?.uri;
    }

    return null;

  } catch (error) {
    console.error('Image generation error:', error.message);
    return null;
  }
}

/**
 * Generate all images for a proposal
 */
async generateProposalImages(businessInfo) {
  const { business_name, industry_type, brand_colors } = businessInfo;

  const images = {};

  // Hero banner
  images.hero = await this.generateImage(
    `Professional hero banner for a payment processing proposal for a ${industr
    'corporate photography'
  );

  // Savings graphic
  images.savings = await this.generateImage(
      `Abstract visualization of business savings and growth. Upward trending ele
      'infographic'
    );

    // Equipment/technology visual
    images.equipment = await this.generateImage(
      `Modern payment terminal on clean retail counter. Professional product phot
      'product photography'
    );

    return images;
  }
}

module.exports = { ImageGenerator };
```

STEP 5: Update the Main Proposal Generation Flow

Update your main proposal generation function to use the new renderer:

```
// In your main proposal generation file

const { HtmlRenderer } = require('./renderers/html-renderer');
const { ImageGenerator } = require('./services/image-generator');
const { scrapeBusinessLogo } = require('./services/logo-scraper');
const { parseSpreadsheets } = require('./parsers/spreadsheet-parser');

const htmlRenderer = new HtmlRenderer();
const imageGenerator = new ImageGenerator();

async function generateProposal(formData, files) {
  console.log('=== PROPOSAL GENERATION STARTED ===');

  // STEP 1: Parse spreadsheets (your existing logic)
  console.log('Step 1: Parsing spreadsheets...');
  const pricingData = await parseSpreadsheets(files.dp, files.icp);
  console.log('Extracted pricing data:', JSON.stringify(pricingData, null, 2));

  // Validate extraction
  if (!pricingData.current_monthly_volume || pricingData.current_monthly_volume =
    throw new Error('Spreadsheet parsing failed: Could not extract monthly volume
  }

  // STEP 2: Scrape merchant logo (if website provided)
```

```
console.log('Step 2: Scraping merchant logo...');
let merchantLogo = null;
if (formData.business_website) {
  merchantLogo = await scrapeBusinessLogo(formData.business_website);
  console.log('Logo found:', merchantLogo ? 'Yes' : 'No');
}

// STEP 3: Generate images with Gemini (optional)
console.log('Step 3: Generating images...');
let images = {};
try {
  images = await imageGenerator.generateProposalImages({
    business_name: formData.business_name,
    industry_type: formData.industry_type || 'retail'
  });
} catch (err) {
  console.log('Image generation skipped:', err.message);
}

// STEP 4: Combine all data for template
console.log('Step 4: Preparing template data...');
const proposalData = {
  // Form data (agent + merchant)
  ...formData,

  // Pricing data from spreadsheets
  ...pricingData,

  // Calculated fields
  current_avg_ticket: pricingData.current_monthly_volume / pricingData.current_
  current_annual_fees: pricingData.current_monthly_fees * 12,
  dp_annual_cost: pricingData.dp_monthly_cost * 12,
  icp_annual_cost: pricingData.icp_monthly_cost * 12,

  // Images
  merchant_logo_url: merchantLogo,
  hero_image_url: images.hero,

  // Recommendation logic
  recommended_option: pricingData.dp_monthly_savings > pricingData.icp_monthly_
    ? 'Dual Pricing'
    : 'Interchange Plus',

  // Business description (from research or default)
  business_description: formData.business_description ||
    `${formData.business_name} is a ${formData.industry_type || 'local business
    opportunity_statement: `By optimizing your payment processing, you could save
      Math.max(pricingData.dp_annual_savings, pricingData.icp_annual_savings).toL
    } annually while improving the customer experience.`
  };

  // STEP 5: Generate PDF
  console.log('Step 5: Rendering PDF...');
  const outputPath = `/outputs/proposals/${formData.business_name.replace(/\s+/g,

  const result = await htmlRenderer.generateProposal(proposalData, outputPath);

  console.log('=== PROPOSAL GENERATION COMPLETE ===');
  console.log('Output:', outputPath);

  return result;
}

module.exports = { generateProposal };
```

STEP 6: Install Dependencies

```
npm install handlebars puppeteer
```

If Puppeteer doesn’t work in Replit’s environment, use this alternative:

```
npm install html-pdf-node
```

And update the PDF generation:

```
const html_to_pdf = require('html-pdf-node');

async function htmlToPdf(htmlContent, outputPath) {
  const options = { format: 'Letter', printBackground: true };
  const file = { content: htmlContent };

  const pdfBuffer = await html_to_pdf.generatePdf(file, options);
  fs.writeFileSync(outputPath, pdfBuffer);

  return outputPath;
}
```

WHAT THIS UPGRADE DOES

Before	After
Plain tables	Styled cards with gradients and shadows
Black & white	Full color with PCBancard brand palette
No images	Gemini-generated hero banner + graphics
Generic layout	Professional 4-page document design
Static text	Dynamic highlighting (winner badges, color-coded savings)
Basic formatting	Modern CSS (rounded corners, spacing, typography)

TESTING CHECKLIST

After implementing, verify:

```
□ HTML template renders without errors
□ All placeholders are replaced with real data
□ Merchant logo appears (when website provided)
□ Gemini images appear (when API key configured)
□ Numbers are formatted correctly (currency, percentages)
□ PDF generates without Puppeteer errors
□ 4 pages render properly with page breaks
□ Colors print correctly (printBackground: true)
□ Fallbacks work when images/logo unavailable
```

GRADUAL ROLLOUT

To avoid breaking existing functionality:

1. **Add new files** (don't modify existing ones initially)
2. **Add a feature flag**: `USE_VISUAL_RENDERER=true`
3. **Test with flag enabled** on a few proposals
4. **Once stable**, make it the default

```
const useVisualRenderer = process.env.USE_VISUAL_RENDERER === 'true';

if (useVisualRenderer) {
  return await htmlRenderer.generateProposal(data, outputPath);
} else {
  return await existingRenderer.generateProposal(data, outputPath);
}
```

This way you can toggle back instantly if issues arise.