

# Replit Implementation Prompt: Dual Renderer Setup

Copy this into Replit to implement both rendering options.

---

## PROMPT START

Add dual rendering capability to the PCBancard Proposal Generator. The agent should be able to choose between:

1. **Replit Native** - Generate DOCX/PDF locally using docx-js
2. **Gamma API** - Generate polished presentations via Gamma's API

## STEP 1: Add Environment Variable


The Gamma API key has been uploaded. Add to your .env:

```
GAMMA_API_KEY=sk-gamma-xxxxxxx
```


## STEP 2: Create Renderer Selection UI

Add a toggle/selector to the proposal form that lets the agent choose:

OUTPUT METHOD

 Replit  
Native

- Fast
- Editable
- DOCX/PDF

 Gamma  
Presentation

- Beautiful
- Animated
- PDF/PPTX

Format: [PDF ▼]

## STEP 3: Implement Gamma API Integration

Create a Gamma renderer that:

1. Takes the Claude-generated proposal blueprint
2. Formats it as structured text with `\n---\n` card breaks
3. POSTs to `https://public-api.gamma.app/v1.0/generations`
4. Polls `GET /generations/{id}` until status is "completed"
5. Downloads the PDF/PPTX from the returned URL
6. Stores locally and returns download link to agent

### Key API details:

- Header: `X-API-KEY: sk-gamma-xxxxxxx` (NOT Bearer token)
- Include `"exportAs": "pdf"` or `"pptx"` to get downloadable file
- Download URL expires - save the file immediately when generation completes

### Sample POST body:

```
{
  "inputText": "# Slide 1 Title\nContent...\n\n---\n\n# Slide 2 Title\nContent...",
  "format": "presentation",
  "numCards": 8,
  "themeId": "Professional",
  "exportAs": "pdf",
  "cardOptions": {
    "dimensions": "16x9"
  },
  "contentOptions": {
    "textDensity": "medium"
  }
}
```

## STEP 4: Add Fallback Logic

If Gamma fails (API down, out of credits, timeout), automatically fall back to Replit Native:

```
try {
  if (selectedRenderer === 'gamma') {
    result = await gammaRenderer.generate(blueprint, options);
```

```

    } else {
      result = await nativeRenderer.generate(blueprint, options);
    }
  } catch (error) {
    console.warn(`${selectedRenderer} failed, falling back...`);

    // Try the other renderer
    if (selectedRenderer === 'gamma') {
      result = await nativeRenderer.generate(blueprint, options);
      result.fallback = true;
      result.fallbackReason = error.message;
    }
  }
}

```

## STEP 5: Update Results Display

Show which renderer was used and provide appropriate download options:

### For Gamma:

- Download PDF button
- Download PPTX button (if requested)
- “Open in Gamma” link (for editing)
- Show credits remaining

### For Replit Native:



- Download PDF button
- Download DOCX button
- No external links needed

### If fallback occurred:



- Show message: “Gamma was unavailable, generated with Replit Native instead”

## STEP 6: Add Status Indicator

Show renderer availability in the UI:

Renderers:  Replit Native  Gamma (847 credits)

or if Gamma is down:

Renderers:  Replit Native  Gamma (unavailable)

Check Gamma status on page load by calling `GET /themes` endpoint.

## STEP 7: Test Both Paths

Create test buttons in dev mode:

```
[Test Replit Native] [Test Gamma API] [Test Fallback]
```

Verify:

- Both renderers produce complete proposals
- All numbers match the input data
- PCBancard branding appears (logo, colors)
- Downloads work correctly
- Fallback triggers when Gamma is disabled

## EXPECTED BEHAVIOR

| Agent Action                | Result   |
|-----------------------------|--|
| Selects Replit + PDF        | DOCX generated, converted to PDF, download link shown                            |
| Selects Replit + DOCX       | DOCX generated directly, download link shown                                     |
| Selects Gamma + PDF         | Gamma generates presentation, PDF downloaded, both Gamma link and download shown |
| Selects Gamma + PPTX        | Same as above but PPTX file  |
| Selects Gamma but API fails | Automatic fallback to Replit Native with notification                            |

## PROMPT END

---

## Quick Test Checklist

After implementation, verify:

- ☐ `Renderer toggle appears in form`
- ☐ `Selecting Replit shows PDF/DOCX format options`
- ☐ `Selecting Gamma shows PDF/PPTX/Link options`
- ☐ `Replit Native generates successfully`
- ☐ `Gamma API connects (check console for auth errors)`
- ☐ `Gamma generation completes (may take 30+ seconds)`
- ☐ `Gamma file downloads to server`
- ☐ `Download links work for agent`
- ☐ `Fallback triggers when Gamma is disabled`
- ☐ `Credits remaining displays after Gamma generation`