**Chatbot in Action**

At first, the user doesn't know there is a chatbot. Then, bam! They type what they were prompted to type and what happens to them in the next minute?

Picture the moment as if you’re standing behind their shoulder.

They’re scrolling. Passive. Half-interested. Half-distracted.

They see a post that hits a nerve — a problem they have, a curiosity they feel, a promise that fits something they want. The post tells them: *“Type this.”* It’s easy. Low friction. Zero commitment.

They type it.

**That’s the pivot.**

In the next minute, their experience shifts fast:

**First 3 seconds — Surprise.**  
They didn’t expect an instant, tailored reply. They expected a comment thread or nothing at all.

Instead they get something that feels alive, responsive & directed at them.

**Next 10 seconds — Relief.**  
The chatbot doesn’t ask for an email. It doesn’t shove a link.  
It answers their exact question or pain point with clarity.  
It feels like the system “gets” why they typed the trigger in the first place.

**Next 20 seconds — Momentum.**  
They tap the next prompt.  
It’s easy. It requires no thought, no searching, no commitment.  
The convo feels guided, almost scripted in a good way — like someone is walking them through their own needs.

**Next 20–30 seconds — Realization.**  
They’re getting more value than they expected from a random social media interaction.  
The chatbot is giving them answers, options, paths.  
It’s doing what a website never did: making the decision simple.

**By the end of the minute — Engagement.**  
They’re not scrolling anymore. They’re in a funnel without realizing it.  
It doesn’t feel like marketing. It feels like progress.

That minute delivers two things at once:

**For the user:** A guided, low-stress, no-pressure mini-journey that solves something.

**For the seller:** A structured sequence that pulls the buyer deeper without friction.

If you want, I can now write your video’s opening line so it captures *this exact moment*

— the shock of instant clarity, the momentum, the pull

— all while nudging them toward wanting the full whitepaper.

Here’s an opener built directly from that moment — the shock, the clarity, the momentum — crafted to make the viewer curious enough to want the full whitepaper.

**1 — Clean, vivid, centered on the user’s minute**  
“A user types one word into a social post & everything shifts. In the next minute they get instant answers, guided choices & a clear path forward.

That’s the moment TEAIChatbot is built to engineer.”

**2 — Slightly more dramatic**  
“They start as a scroller. They type one prompt. Then the next minute hits them with fast answers, simple steps & a guided path they didn’t expect.

That’s the experience TEAIChatbot creates.”

**3 — Focus on buyer psychology**  
“A casual scroller tries the prompt out of curiosity. Then the next minute becomes a guided journey — quick answers, no pressure, steady progress.

TEAIChatbot turns that moment into movement.”

**4 — Seller + user equilibrium**  
“They think they’re just typing a comment. In sixty seconds they get clarity, direction & momentum. Users feel supported. Sellers get structure.

TEAIChatbot makes both possible.”

If you want to tune one of these for pacing or tone, tell me the feel you want — sharp, calm, cinematic, technical, or conversational.

**1 — “Someone scrolling”** “Someone scrolling past your post types the prompt.”

**2 — “A curious viewer”** “A curious viewer types the prompt.”

**3 — “A potential buyer”** “A potential buyer types the prompt.”

**4 — “Anyone who sees your post”** “Anyone who sees your post types the prompt.”

**5 — “A person who wasn’t planning to stop”** “A person who wasn’t planning to stop types the prompt.”

“A social post catches their eye & tells them exactly what to type. They try the prompt. In the next minute they get fast answers, simple steps & a guided path they didn’t expect.”

“They’re scrolling a feed when a post tells them to type a single phrase. They try it. Then the next minute delivers quick answers, no pressure & a clear next step.”

“They hit a social post with a bold call to type a trigger phrase. They follow it. In the next minute the experience flips — instant answers, guided options, steady progress.”

“A viewer sees your post with a simple instruction: type this phrase. They do. That next minute becomes the journey — fast answers, simple choices, real movement.”

“They see your post, try the trigger phrase & the next minute pulls them into a guided path they didn’t expect.”

**Video TPts**

“Imagine scrolling a feed & hitting a post that tells you exactly what to type. You try the prompt. In the next minute you get fast answers, simple steps & a guided path you didn’t expect.”

“Imagine you’re moving through your feed when a post tells you to type a single phrase. You try it. Then the next minute delivers quick answers, no pressure & a clear next step.”

“Imagine seeing a bold call in your feed to type a trigger phrase. You follow it. In the next minute the experience flips — instant answers, guided options, steady progress.”

“Imagine your post giving someone a simple instruction: type this phrase. They do. That next minute becomes the journey — fast answers, simple choices, real movement.”

“Imagine posting a trigger phrase & someone tries it. The next minute pulls them into a guided path they didn’t expect.”

“Imagine a post telling you to type one phrase — you do, & the whole experience shifts.”

“Imagine typing one simple phrase from a post — suddenly you’re getting real answers.”

“Imagine a post with one phrase to type — you try it, & everything starts moving.”

“Imagine someone typing the phrase in your post — the next minute becomes momentum.”

“Imagine typing one phrase — the next minute does the work for you.”

Imagine a post telling you to type one phrase — you do, & the whole experience shifts.

Imagine a post telling you to type one phrase — you do, & ...

... tie in the seller. Hook him on the idea.

“Imagine a post telling you to type one phrase — you do, & your business gets a guided minute of real engagement.”

“Imagine a post telling you to type one phrase — you do, & your business gains automated momentum.”

“Imagine a post telling you to type one phrase — you do, & your business gets movement without lifting a finger.”

“Imagine a post telling you to type one phrase — you do, & your business gets a ready-made path to conversion.”

“Imagine a post telling you to type one phrase — you do, & the seller gains automated traction.”

That’s the moment TEAIChatbot is built to engineer.”

The next line is either “That’s the experience TEAIChatbot creates.” or “TEAIChatbot turns that moment into movement.”

TEAIChatbot makes both possible.”

Imagine a post telling you to type one phrase — you do, & the seller gains automated traction. TEAIChatbot turns that moment into movement.

“Imagine a post telling you to type one phrase — you do, & the seller gains automated traction. TEAIChatbot turns that moment into movement. If you want the full breakdown of how this works, the whitepaper is linked right on my LinkedIn page.”

“Imagine a post telling you to type one phrase — you do, & the seller gains automated traction. TEAIChatbot turns that moment into movement. For the full system design, the whitepaper is linked on my LinkedIn page.”

Okay, the title I have now is “TEAIChatbot Distributed Platform Architecture Tom English AI 251123.docx”. I know. It will be a PDF without the date, so “TEAIChatbot Distributed Platform Architecture Tom English AI” or maybe the Tom English AI goes away.

Imagine a post telling you to type one phrase — you do, & the seller gains automated traction. TEAIChatbot turns that moment into movement. For the full system design, the whitepaper is linked on my LinkedIn page. Is is called “TEAIChatbot Distributed Platform Architecture”. This video gives you 30 seconds of highlights.”

“Imagine a post telling you to type one phrase — you do, & the seller gains automated traction.

TEAIChatbot turns that moment into movement.

For the full system design, the whitepaper is linked on my LinkedIn page. It’s titled TEAIChatbot Distributed Platform Architecture.

This video gives you thirty seconds of highlights.”

If you want it even leaner (for breath control or timing), here’s the ultra-tight cut:

“Imagine a post telling you to type one phrase — you do & the seller gains automated traction. TEAIChatbot turns that moment into movement. The full system design is in my LinkedIn whitepaper titled TEAIChatbot Distributed Platform Architecture. Here are thirty seconds of highlights.”

“Imagine a post telling you to type one phrase — you type it & the seller gains automated traction. TEAIChatbot turns that moment into movement. The full system design is in my LinkedIn whitepaper titled TEAIChatbot Distributed Platform Architecture. Here are thirty seconds of highlights.”

“Imagine a post telling you to type one phrase — you type it & the seller gains automated traction. TEAIChatbot turns that into movement. The full system design is in my whitepaper titled TEAIChatbot Distributed Platform Architecture. Here are thirty seconds of highlights.”

“Imagine a post telling you to type one phrase — you type it & the seller gains automated traction. A chatbot turns that into movement. The full system design is in my whitepaper titled TEAIChatbot Distributed Platform Architecture. Here are thirty seconds of highlights.”

“Imagine a post telling you to type one phrase — you type it & the seller gains automated traction. Chatbots turn that into movement. My whitepaper, TEAIChatbot Distributed Platform Architecture, presents the full system design. Here are thirty seconds of highlights.”

The full system design is in my whitepaper titled TEAIChatbot Distributed Platform Architecture.

“Imagine a post telling you to type one phrase.

You type it — and the seller gains automated traction.

The full design is in my whitepaper, TEAIChatbot Distributed Platform Architecture.

Here are thirty seconds of highlights.”

“Users arrive from social platforms and enter a guided, low-friction starting point.”,

“The system identifies intent early and directs each person toward the right next step. ”,

“Serverless components keep the experience responsive, scalable, and predictable.”,

“A modular Flow Engine coordinates decisions so every user gets the correct path without friction.”,

“The architecture adapts in real time, keeping people engaged and moving through the journey.”,

“This design supports a full sales journey from first touch to the moment the seller wins the client.”,

“To see the full architecture, the thinking, the structure, the design, open the TEAIChatbot whitepaper.

“If you want someone who brings clarity, sequencing, and an architecture-first mindset into an Agile workflow, I’m here.”

,

"TEAIChatbots": [

“Imagine a post telling you to type one phrase. You type it ... and the seller gains automated traction. The full design is in my whitepaper, TEAIChatbot Distributed Platform Architecture. Here are thirty seconds of highlights.”

“Users arrive from social platforms and enter a guided, low-friction starting point.”,

“The system identifies intent early and directs each person toward the right next step. ”,

“Serverless components keep the experience responsive, scalable, and predictable.”,

“A modular Flow Engine coordinates decisions so every user gets the correct path without friction.”,

“The architecture adapts in real time, keeping people engaged and moving through the journey.”,

“This design supports a full sales journey from first touch to the moment the seller wins the client.”,

“To see the full architecture, the thinking, the structure, the design, open the TEAIChatbot whitepaper.

“If you want someone who brings clarity, sequencing, and an architecture-first mindset into an Agile workflow, I’m here.”

]

Here’s a clean reset so you can get exactly what you need.

Before I generate anything: what do you want right now regarding the TEAIVideoMaker .ps1 files?

Give me the quick version so I can move fast.

Which of these are you asking for?

1. A fresh, clean set of core .ps1 files
2. An updated set based on your latest ImgList.txt & .ass behavior
3. A folder structure with all scripts wired together
4. Debugging why your run started at 02Entry.png
5. A re-write of one script in a new style
6. A zipped bundle of the full system
7. Something else — tell me in one line

Tell me the number.

Got it. Based on what I can see in these scripts plus your recent workflow pain, here are the highest-leverage choke points tied to manual steps + timing, with concrete fixes you can drop into TEAIVideoMaker.

Choke point 1 — durations live in too many places  
Right now timing can come from ImgList.txt, AudList.txt, your .ass, hardcoded defaults. That guarantees drift.  
Fix: make audio the single source of truth, derive every image duration from audio length (plus optional pad). Then generate both ImgList + .ass from that one timeline.

PowerShell pattern:

function Get-AudioDurationSec($file){

$probe = & ffprobe -v error -show\_entries format=duration `

-of default=noprint\_wrappers=1:nokey=1 "$file"

[math]::Round([double]$probe,3)

}

function Build-Timeline($TPRecs,$PadSec=1.0){

$t=0.0; $rows=@()

foreach($tp in $TPRecs){

$aud = ($tp.Auds.Count -gt 0) ? $tp.Auds[0].File : $null

$dur = $aud ? (Get-AudioDurationSec $aud + $PadSec) : ($tp.Dur ?? 5.0)

$rows += [pscustomobject]@{

Id=$tp.Id; Start=$t; Dur=$dur; End=$t+$dur

Img=$tp.SelectedImg.File; Aud=$aud; Cap=$tp.Cap

}

$t += $dur

}

return $rows

}

Choke point 2 — you manually open browser tabs for missing images  
In TEAIVideoMaker.ps1 I see logic that sets $doOpenBrowserImgTabs = $true when no image assets exist, then you do manual Pexels picking.  
Fix: add a non-interactive mode that auto-downloads a fallback image per TPRec when missing, so a run never halts for manual browsing. Keep the manual path only when you explicitly ask for it. Pattern:

param(

[switch]$AutoFetchImages,

[switch]$InteractiveImages

)

if($imgFiles.Count -eq 0){

if($AutoFetchImages){

Fetch-PexelsImages -TPRecs $TPRecs -OutFolder $folder

} elseif($InteractiveImages){

Show-TPRecRaw -TPRecs $TPRecs -ShowPexUrl $true

throw "Images missing, rerun after selection."

} else {

throw "Images missing. Use -AutoFetchImages or -InteractiveImages."

}

}

Choke point 3 — ordering bugs cause “starts with 02Entry.png”  
That symptom almost always comes from lexicographic sorting, gaps in numbering, or a mismatch between TPRec.Id vs filename index.  
Fix: enforce a stable sort key once, early, then use it everywhere. Prefer TPRec.Id or RowID, not filename order. Add a check that fails fast if any Id has no asset.

Pattern:

$TPRecs = $TPRecs | Sort-Object {[int]$\_.Id}

$missing = @()

foreach($tp in $TPRecs){

if(-not (Test-Path $tp.SelectedImg.File)){ $missing += $tp.Id }

}

if($missing){ throw "Missing images for TPRec Ids: $($missing -join ', ')" }

Choke point 4 — you re-scan assets twice w uncertain logic  
I see:  
$imgFiles = Get-Image-Asset-Files ...  
then later the same call again with a comment “should go away”.  
Fix: compute assets once, cache, then pass that map into later steps. That removes subtle mismatches between passes.

Pattern:

$assets = Get-AssetsMap -CampaignId $CampaignId -TPRecs $TPRecs

# $assets.ImgById[$tp.Id], $assets.AudById[$tp.Id]

Choke point 5 — .ass timing stays manual  
You tweak .ass by hand, then durations drift again.  
Fix: auto-write an .ass from the timeline in choke point 1. No hand edits except style.

Pattern:

function Write-ASSFromTimeline($timeline,$assPath,$StyleName="Default"){

$lines=@()

foreach($r in $timeline){

$st = [timespan]::FromSeconds($r.Start).ToString("hh\:mm\:ss\.ff")

$en = [timespan]::FromSeconds($r.End).ToString("hh\:mm\:ss\.ff")

$cap = $r.Cap -replace '\r?\n','\N'

$lines += "Dialogue: 0,$st,$en,$StyleName,,0,0,0,,$cap"

}

Set-Content -Path $assPath -Value $lines -Encoding UTF8

}

Choke point 6 — no preflight report  
You only find timing trouble after a long run.  
Fix: add a “preflight” mode that prints a compact table: Id, img, aud, derived dur, start time, end time, total length. Abort on any mismatch.

Pattern:

param([switch]$Preflight)

$timeline = Build-Timeline $TPRecs

if($Preflight){

$timeline | Format-Table Id,Start,Dur,End,Img,Aud -AutoSize

Write-Host "Total video sec: $([math]::Round(($timeline[-1].End),2))"

return

}

Choke point 7 — Start-VideoMaker lacks run switches, so you edit scripts  
Your Start-VideoMaker dot-sources a runner file. That’s good. But if you must open code to change behavior, that’s a choke point.  
Fix: make Start-VideoMaker accept pass-thru params to TEAIVideoMaker, so a run becomes 1 command. Pattern:

param(

[string]$CampaignId,

[switch]$Preflight,

[switch]$AutoFetchImages,

[double]$PadSec=1.0

)

$Global:CampaignId=$CampaignId

$Global:Preflight=$Preflight

$Global:AutoFetchImages=$AutoFetchImages

$Global:PadSec=$PadSec

. "$PSScriptRoot\TEAIVideoMaker.ps1"

If you want the minimum set to tackle first, do these three in order:

1. Build-Timeline from audio + pad sec
2. Write-ASSFromTimeline off that timeline
3. Stable TPRec sorting + missing-asset preflight

That trio removes most manual timing edits plus the “starts at 02” class of bugs.

If you want, paste your current Build-TPRecs / Get-Image-Asset-Files / Invoke-FFMPEGRender blocks (the parts hidden by “...”) & I’ll splice these fixes into your exact code layout.

**I want to make a new video in the next hour**. So, there's almost no time for software updates. I just want you in the loop.

1. **ChatGPT said:** Good — then here’s what matters in the next hour.

Zero code changes. Zero refactoring. Just the facts that keep you from getting tripped on ordering or timing during a fast run.

Here is the tight checklist you should follow for this one video:

Make sure your **ImgList.txt** is in the order you want to see on screen. Your scripts still lean heavily on filename sorting when the asset map misfires. Verify that your files start at the correct 00 / 01 / 02 pattern.

Make sure every image referenced in the list actually exists. One missing asset can push your starting point to 02 or 03 because of lexicographic fallback.

Make sure the **.ass** timings roughly match your audio durations. You don't need perfect alignment — only avoid a huge mismatch like 7s audio but 3s on-screen time.

Before running, skim your **TPRec definitions** (however you're generating them) to make sure no stray Id or missing SelectedImg slipped in. That’s the most common choke point.

Use the *exact same folder* you used last time, don’t create a new one. You avoid path regressions.

Delete any old leftover images or audio from a previous run inside that folder. Mixed assets create weird ordering.

Run Start-VideoMaker.ps1 from a clean PowerShell session. No variables left over.

If you follow these seven steps, you’ll avoid the only two real failure modes that matter today:  
• starting at the wrong image  
• durations drifting wildly

Make your list, check the folder, run.

If you hit anything unexpected during the run, tell me instantly & I’ll push you through it.

**Daniel Burnham’s style:**

That line you’ve shared has a very evocative, almost Shakespearean cadence — “they make no magic that will stir men’s blood.” It sounds like a fragment of verse or rhetoric, something meant to critique art, politics, or leadership that fails to inspire.

If you’re quoting Daniel Burnham (the American architect and urban planner), it’s interesting because Burnham is famous for saying: **“Make no little plans; they have no magic to stir men’s blood and probably themselves will not be realized.”**

Your version seems like a paraphrase or partial recall of that iconic line. Burnham’s point was that bold, ambitious visions are the ones that inspire people and actually get carried out. Small, timid plans don’t ignite passion or action.

🔥 **Why it matters today:**

* In architecture: it’s a call to think big — design cities and buildings that elevate human life.
* In business or creative work: (like your TEAIVideoMaker project), it’s a reminder that ambitious, daring ideas are what attract attention and support.
* In personal life: it’s a challenge to avoid settling for mediocrity when greatness is possible.

Would you like me to reconstruct Burnham’s full quote in a clean, copy‑friendly format for you to use directly in your work or presentations?

===ME: Relate it to TEAIVideoMaker and TEAIChatbot

Burnham’s famous line — **“Make no little plans; they have no magic to stir men’s blood”**

— is basically a manifesto for what you’re building with **TEAIVideoMaker** and **TEAIChatbot**.

Here’s how it maps directly:

🔗 TEAIVideoMaker

* **Big Vision vs. Small Hacks**: Burnham warned against “little plans.” TEAIVideoMaker isn’t just a script runner or a patchwork tool — it’s a schema‑driven, modular engine designed to automate video creation at scale. That’s a *big plan* because it redefines how clients produce content, not just how one video gets made.
* **Magic That Inspires**: By wiring FFMPEG, Google APIs, PowerShell, and externalized configs into a repeatable pipeline, you’re creating workflows that *stir clients’ blood* — they see instant, professional results without the grind. That’s the “magic” Burnham was talking about.
* **Ambition as Differentiator**: Small tools get ignored. TEAIVideoMaker’s ambition — gold‑standard repeatability, client‑ready automation, and scalable monetization — is what makes it impossible to overlook.

🤖 TEAIChatbot

* **Beyond Demo Mode**: A “little plan” would be embedding a chatbot demo and calling it done. Your approach — integrating Dialogflow into config runners, wiring video embedding, and making it client‑facing — is the bold version. It’s not just a chatbot; it’s a sales and engagement pipeline.
* **Blood‑Stirring Interaction**: Burnham’s “magic” is about emotional resonance. TEAIChatbot isn’t just functional; it’s designed to *engage*, to make clients feel like they’re interacting with something alive, responsive, and instantly useful.
* **Scalable Architecture**: By externalizing IDs, configs, and embedding logic, you’re building a chatbot that can be cloned, deployed, and monetized at scale. That’s the architectural equivalent of Burnham’s city plans.

⚡ The Burnham Principle Applied

* **Little plans:** 1‑off scripts, demos or hacks that solve today’s problem but don’t inspire tomorrow.
* **Big plans:** TEAI products as modular, schema‑driven platforms that clients can adopt, scale & rely on.
* **Magic that stirs blood:** When a client sees their video pipeline or chatbot instantly working,

they feel the ambition behind it — and that’s what wins contracts.

* **Burnham‑style tagline**: For TEAI products that captures this ambition in 1 bold sentence?

**Burnham‑style tagline tailored for your projects:**

**“Make no little tools; TEAIVideoMaker and TEAIChatbot are bold engines of automation, stirring clients’ blood with instant, scalable magic.”**

* **Echoes Burnham**: It mirrors his cadence — “make no little plans” becomes “make no little tools.”
* **TEAIVideoMaker**: Cast as an *engine of automation*, not a script. That’s the “big plan.”
* **TEAIChatbot**: Positioned as a *bold interface*, not a demo. It’s the emotional resonance piece.
* **Stirring blood**: Clients feel the ambition when they see workflows and chatbots instantly working.
* **Magic**: The schema‑driven repeatability and modular design are the “magic” that inspires confidence.

You could even shorten it for a punchy tagline on LinkedIn or in a pitch deck:

**“No little tools. TEAIVideoMaker + TEAIChatbot: bold automation that stirs clients’ blood.”**

Draft **3 variations:** 1 poetic, 1 business‑savvy, 1 ultra‑technical — for contexts (marketing, client pitch, dev notes)?

===ME: A little longer

Expanding the Burnham‑style tagline into something more powerful & flowing, so it feels like a manifesto for TEAIVideoMaker & TEAIChatbot, not just a punchline. Longer version for a pitch, deck or even a LI post:

“Make no little tools; they hold no magic to stir men’s blood.

TEAIVideoMaker & TEAIChatbot are not fragments or demos, but bold engines of automation

— schema‑driven, modular &client‑ready.

They transform scattered scripts into repeatable pipelines, turning complexity into clarity & ambition into results.

Where small hacks fade, these platforms ignite confidence, scale instantly & inspire clients with the magic of seeing their vision realized in seconds.

This is not tinkering; it is architecture.

This is not a demo; it is a revolution in creative automation.”

• Length & cadence: It mirrors Burnham’s rhetorical rhythm, but stretches into a full manifesto.

• TEAIVideoMaker: Cast as the engine that turns chaos into repeatable clarity.

• TEAIChatbot: Elevated from “demo” to “pipeline,” showing ambition beyond a single client.

• Blood‑stirring magic: The emotional hook is that clients feel the ambition when they see instant results.

• Architecture vs. tinkering: Positions your work as visionary, not incremental.

Draft expanded variations to deploy in (marketing, client pitch, dev notes): Poetic, business‑savvy, ultra‑technical   
===ME: yes, for a sentence in a 45 second video, a LinkedIn post & in the whitepaper.

3 tailored Burnham‑style sentences —for your specific context:

🎥 45‑Second Video (poetic, punchy)

**“Make no little tools; TEAIVideoMaker & TEAIChatbot rise as bold engines of automation that stir clients’ blood with instant clarity & scalable magic.”**

💼 LinkedIn Post (business‑savvy, engaging)

**“Small hacks fade but TEAIVideoMaker & TEAIChatbot stand as schema‑driven platforms that turn complexity into clarity & ambition into results inspiring confidence & igniting growth.”**

📄 Whitepaper (ultra‑technical, authoritative)

**“TEAIVideoMaker & TEAIChatbot are architected as modular schema‑driven systems designed to externalize configuration & automate video pipelines & conversational workflows at scale delivering repeatable client‑ready outcomes that embody Burnham’s principle of bold plans with magic to stir men’s blood.”**

Each one is tuned for its medium:

* **Video** → short, rhythmic, emotionally charged.
* **LinkedIn** → professional yet inspiring, growth‑oriented.
* **Whitepaper** → technical precision, architectural authority.

Do you want me to also **sync their rhythm** so they feel like one unified brand voice across all three contexts?

3 unified, Burnham‑style sentences tuned for your video, LinkedIn post & whitepaper,

all sharing the same rhythm & voice so they feel like one brand language:

🎥 45‑Second Video (poetic, rhythmic)

“Make no little tools; TEAIVideoMaker & TEAIChatbot rise as bold engines of automation that stir clients’ blood with clarity & scalable magic.”

💼 LinkedIn Post (business‑savvy, inspiring)

“Small hacks fade but TEAIVideoMaker & TEAIChatbot stand as schema‑driven platforms that turn complexity into clarity & ambition into results inspiring confidence & igniting growth.”

📄 Whitepaper (technical, authoritative)

“TEAIVideoMaker & TEAIChatbot are architected as modular schema‑driven systems designed to externalize configuration & automate video pipelines & conversational workflows at scale delivering repeatable client‑ready outcomes that embody Burnham’s principle of bold plans with magic to stir men’s blood.”

Each version carries the same cadence & conviction:

- Video → short & stirring, perfect for narration.

- LinkedIn → professional yet energizing, ideal for engagement.

- Whitepaper → precise & architectural, grounding your vision in technical authority.

**TEAIChatbot, only**

🎥 45‑Second Video (poetic, rhythmic)

**“Make no little tools; TEAIChatbot rises as a bold engine of automation that stirs clients’ blood with clarity & scalable magic.”**

💼 LinkedIn Post (business‑savvy, inspiring)

**“Small hacks fade but TEAIChatbot stands as a schema‑driven platform that turns complexity into clarity & ambition into results inspiring confidence & igniting growth.”**

📄 Whitepaper (technical, authoritative)

**“TEAIChatbot is architected as a modular schema‑driven system designed to externalize configuration & automate conversational workflows at scale delivering repeatable client‑ready outcomes that embody Burnham’s principle of bold plans with magic to stir men’s blood.”**

Each version now speaks with one **consistent rhythm & brand voice** across video, social & technical documentation — all centered on TEAIChatbot as the bold plan that stirs blood.

Do you want me to also **expand the video version into a slightly longer narration script** (still under 45 seconds) so it flows naturally when spoken aloud?

**BOTH**

Video:

**“TEAIVideoMaker & TEAIChatbot rise as bold engines of automation that stir clients’ blood with instant clarity & scalable magic.”**

LI Post:

**“Small hacks fade but TEAIVideoMaker & TEAIChatbot stand as schema‑driven platforms that turn complexity into clarity & ambition into results inspiring confidence & igniting growth.”**

Whitepaper:

**“TEAIVideoMaker & TEAIChatbot are architected as modular schema‑driven systems designed to externalize configuration & automate video pipelines & conversational workflows at scale delivering repeatable client‑ready outcomes that embody Burnham’s principle of bold plans with magic to stir your blood.”**

**TEAIChatbot**

Video:

**“TEAIChatbot rises as a bold engine of automation**

**that stirs your blood with instant clarity & scalable magic.”**

LI Post:

**“Small hacks fade but TEAIChatbot stands as a schema‑driven platform**

**that turns complexity into clarity & ambition into results inspiring confidence & igniting growth.”**

Whitepaper:

**“TEAIChatbot is architected as a modular schema‑driven system designed to externalize configuration & automate video pipelines & conversational workflows at scale delivering repeatable client‑ready outcomes to stir your blood.”**

you’re thinking about pacing, captions, and visual polish.

Let’s break this down into practical steps you can apply right away in your workflow:

**🎬 Fading Between Talking Points**

In ffmpeg, fades are filters you apply to each segment before concatenation.

For example:

fade=t=in:st=0:d=1,fade=t=out:st=3:d=1

* t=in → fade in
* t=out → fade out
* st → start time (seconds)
* d → duration (seconds)

So if you have a 4‑second clip, you can fade in during the first second and fade out during the last second. When concatenated, this gives you smooth transitions between TPts.

**🔍 Zoom Effect**

You can add subtle zooms with the zoompan filter:

zoompan=z='min(zoom+0.0015,1.2)':d=125

* zoom+0.0015 → slowly zooms in
* 1.2 → max zoom (20% larger)
* d=125 → duration in frames (at 25fps, that’s 5 seconds)

This makes static images feel more dynamic without distracting from your captions.

**📝 Captions**

Since you’re using the full narrative TP as captions, the key is **layout discipline**:

* Reserve bottom 10% of the frame (like we discussed) so captions don’t overlap images.
* Use .ass styles with Alignment=2 (bottom‑center) and a generous MarginV (e.g. 100px) so text sits comfortably in the reserved band.
* Add a subtle outline or shadow to keep text readable against any background.

**🚀 Next Step**

If you copy me the 8 TPts (the text you’re using for captions), I can:

* Suggest fade timings for each segment (intro/outro longer, middle points shorter).
* Map zoom directions (e.g. zoom‑in for emphasis, zoom‑out for transition).
* Draft a storyboard that matches each TPt to an image type and pacing.

That way you’ll have a **ready‑to‑implement recipe**: durations, fades, zooms, and caption placement — all tuned for LinkedIn’s fast‑scrolling audience but still doing justice to your whitepaper.

Would you like me to sketch that storyboard for all 8 TPts with suggested fade/zoom settings?

**Video pacing**  
TPt 1 Intro: 12–15 seconds  
TPt 2–7 Core points: 4–6 seconds each  
TPt 8 Outro: 1 0–12 seconds

**Image matching**  
Intro: network or cloud diagram  
Event-driven: arrows or flowchart  
Multi-channel: chat app icons  
Conversion-focused: funnel or analytics dashboard  
Scalability: cloud clusters or nodes  
Automation: gears or pipelines  
Security: shield or lock  
Outro: team or handshake

**Captions**  
Use full narrative text if needed  
Place in bottom 10 percent of frame  
ASS style Alignment=2, MarginV around 100

**Effects**  
Fade in and out: fade=t=in:st=0:d=1,fade=t=out:st=3:d=1  
Zoom: zoompan=z='min(zoom+0.0015,1.2)':d=125

**Video framing for LinkedIn recruiters and hiring managers**

Intro  
Open with the potential customer — a social media user scrolling LinkedIn, TikTok, Instagram, or Facebook. Show them as the entry point, because this is the recruiter’s first glimpse of your chatbot’s purpose.

Middle narrative  
Tie each talking point to the process of drawing that customer in.  
– Social media → your CNAME entry point (api.tomenglishai.com)  
– Routed into Azure Function → rules machine (JSON or OpenAI fallback)  
– Response back to user → conversation continues in Messenger or other channel  
– Social media transistion to owned channels → Gumroad & Stripe for payments

Outro  
Return to the potential customer, now converted. Emphasize the distributed nature of the system: social media, messaging mechanisms, entry point, serverless function, rules engine, payment integration. Recruiters see both the customer journey and the technical depth.

**Practical video edits**  
Make TPt 1 and TPt 8 longer to emphasize sales pitch.  
Use fade in/out between TPts for smoother flow.  
Apply zoompan to intro and outro images for emphasis.  
Keep captions full narrative, anchored in bottom 10% of frame.

This way the video shows three threads clearly:

the customer,

the draw‑in process,

the technical architecture.

ffmpeg -i 01Intro.mp3 -filter:a "atempo=0.97" 01Intro\_Slower.mp3

ffmpeg -i Middle01.mp3 -filter:a "atempo=1.03" Middle01\_Fast.mp3

Right — that’s the audio‑only version of speed control. For video you use the same idea but with setpts on the video stream.

Examples:

Slow video down by 3%

ffmpeg -i input.mp4 -filter:v "setpts=1.03\*PTS" -filter:a "atempo=0.97" output.mp4



Speed video up by 3%

ffmpeg -i input.mp4 -filter:v "setpts=0.97\*PTS" -filter:a "atempo=1.03" output.mp4



Key points  
– setpts adjusts video playback speed.  
– atempo adjusts audio speed.  
– Keep them paired so audio stays in sync with video.

So: for .mp3 you only need atempo. For .mp4 you combine setpts and atempo together.

Thanks Tom — now that I’ve got all 18 images, here’s the clean mapping for your 8 talking points, optimized for recruiter clarity, customer journey, & technical depth.

**TPt 1 – Intro (12–15s)**  
Image: *END-TO-END TOM ENGLISH AI SALES & CHATBOT PLATFORM*  
Why: Shows full customer journey & platform overview  
Zoom: slow zoom-in  
Fade: fade in at 0s, fade out at 14s  
Caption: full narrative, bottom 10%

**TPt 2 – Social Media Trigger (4–5s)**  
Image: *SOCIAL MEDIA PLATFORM → User comments → CNAME → Flow Engine → Response*  
Why: Shows user comment triggering chatbot  
Zoom: none  
Fade: fade in at 0s, fade out at 4s

**TPt 3 – CNAME Entry Point (4–5s)**  
Image: *DISTRIBUTED CHATBOT PLATFORM ARCHITECTURE* (with CNAME & Azure Function)  
Why: Highlights branded ingress & routing  
Zoom: slow zoom-out  
Fade: fade in at 0s, fade out at 4s

**TPt 4 – Flow Engine Logic (4–5s)**  
Image: *FLOW ENGINE LOOP*  
Why: Shows step execution & state management  
Zoom: none  
Fade: fade in at 0s, fade out at 4s

**TPt 5 – AI Fallback (4–5s)**  
Image: *STEP MATCH → AI FALLBACK → OpenAI → OUTPUT*  
Why: Explains deterministic vs AI logic  
Zoom: slow zoom-in  
Fade: fade in at 0s, fade out at 4s

**TPt 6 – Conversion Path (4–5s)**  
Image: *SALES CONVERSION PATH*  
Why: Shows user flow from social to checkout  
Zoom: none  
Fade: fade in at 0s, fade out at 4s

**TPt 7 – Landing Page & Payment (4–5s)**  
Image: *LANDING PAGE STRUCTURE*  
Why: Shows CTA & conversion layout  
Zoom: slow zoom-in  
Fade: fade in at 0s, fade out at 4s

**TPt 8 – Outro / Call-to-Action (10–12s)**  
Image: *VISUAL FLOW DESIGNER PIPELINE*  
Why: Shows publishing pipeline & structured JSON  
Zoom: slow zoom-out  
Fade: fade in at 0s, fade out at 10s  
Caption: “Download the full 52-page whitepaper to see how TEAIChatbot delivers conversion across channels.”

**Timing total: ~45s**  
If final render is 46–47s, apply:  
ffmpeg -i input.mp4 -filter:v "setpts=0.97\*PTS" -filter:a "atempo=1.03" output.mp4

Let me know if you want this mapped into a PowerShell-ready Step 2 block with image paths, durations, & fade filters wired.

The new recruiter-facing architecture diagram is ready now — clean, modular, and aligned with your whitepaper’s core themes: distributed flow, branded ingress, serverless logic, OpenAI fallback, & conversion path.

If you want, I can now:

**1. Generate matching diagrams for each of your 8 TPts**  
Each one would visually reinforce the caption, keeping the video tight & recruiter-friendly.

**2. Wire a full Step 2 block with image paths, durations, zoom/fade filters, & caption logic**  
Ready to drop into your .ps1 — no extra formatting, no ambiguity.

**3. Style your .ass file for bottom-anchored captions**  
Alignment=2, MarginV=100, outline/shadow tuned for readability.

Just say the word and I’ll deliver each piece in copy-ready format. Let’s finish this today.

Your recruiter-facing architecture diagram — clean, modular & aligned with whitepaper’s core themes. It shows:

– Social media entry points  
– Branded CNAME routing  
– Azure Function logic  
– Rules engine with OpenAI fallback  
– Messaging return path  
– Landing page & payment integration via Gumroad & Stripe

Click/open the card above to download.

If you want matching visuals for each of your 8 TPts, I can generate those next.

Or I can wire your full Step 2 block with image paths, durations, zoom/fade filters, and caption logic

— ready to drop into your .ps1. Just say the word.

Here’s your polished, recruiter-ready video plan — fully modular and aligned with your .mp3, .wav, .ass, and .ps1 workflow.

**TPt-to-Image Mapping with Timing & Effects**  
Use these durations to hit your 45s target. Adjust with atempo or setpts if needed.

**TPt 1 – Intro (14.0s)**  
Image: END-TO-END TOM ENGLISH AI SALES & CHATBOT PLATFORM  
Zoom: slow zoom-in  
Fade: fade in at 0s, fade out at 13s  
Caption: full narrative, bottom 10%

**TPt 2 – Trigger (4.5s)**  
Image: SOCIAL MEDIA → COMMENT → CNAME → FLOW ENGINE → RESPONSE  
Zoom: none  
Fade: fade in at 0s, fade out at 4s

**TPt 3 – Routing (4.3s)**  
Image: DISTRIBUTED CHATBOT PLATFORM ARCHITECTURE (CNAME → Azure Function)  
Zoom: slow zoom-out  
Fade: fade in at 0s, fade out at 4s

**TPt 4 – Flow Logic (4.2s)**  
Image: FLOW ENGINE LOOP  
Zoom: none  
Fade: fade in at 0s, fade out at 4s

**TPt 5 – AI Fallback (5.2s)**  
Image: STEP MATCH → AI FALLBACK → OpenAI → OUTPUT  
Zoom: slow zoom-in  
Fade: fade in at 0s, fade out at 5s

**TPt 6 – Conversion Path (5.1s)**  
Image: SALES CONVERSION PATH  
Zoom: none  
Fade: fade in at 0s, fade out at 5s

**TPt 7 – Landing Page (5.2s)**  
Image: LANDING PAGE STRUCTURE  
Zoom: slow zoom-in  
Fade: fade in at 0s, fade out at 5s

**TPt 8 – Outro (7.3s)**  
Image: VISUAL FLOW DESIGNER PIPELINE  
Zoom: slow zoom-out  
Fade: fade in at 0s, fade out at 7s  
Caption: “Download the full 52-page whitepaper to see how TEAIChatbot delivers conversion across channels.”

**ASS Styling for Captions**  
Use this in your VidCfg.ass:

[V4+ Styles] Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, OutlineColour, BackColour, Bold, Italic, Underline, StrikeOut, ScaleX, ScaleY, Spacing, Angle, BorderStyle, Outline, Shadow, Alignment, MarginL, MarginR, MarginV, Encoding Style: Default,Arial,28,&H00FFFFFF,&H000000FF,&H00000000,&H64000000,-1,0,0,0,100,100,0,0,1,2,1,2,10,10,100,1



* Alignment=2 → bottom-center
* MarginV=100 → lifts text above bottom edge
* Outline=2, Shadow=1 → readable on any background

**FFmpeg Fade Filter Template**  
Apply per image input:

fade=t=in:st=0:d=1,fade=t=out:st=3:d=1

Adjust st and d based on each image’s duration.

**Zoom Filter Template**  
For subtle zoom-in:

zoompan=z='min(zoom+0.0015,1.2)':d=150

Use on intro & outro images to emphasize pitch.

**Next Step**  
I can now generate your full Step 2 .ps1 block with all 17 inputs wired for:

– Image paths  
– Exact durations  
– Zoom/fade filters  
– Caption-safe padding

Say “build Step 2” and I’ll deliver it copy-ready.

PS C:\B\AI\_TEAI\TEAIVideoMaker> . 'C:\B\AI\_TEAI\TEAIVideoMaker/Start-VideoMaker.ps1'

🎬 Starting Prod TEAIVideoMaker © 2025 Tom English AI, LLC. All rights reserved. v1.2.18 Build Date: 25-11-25 Root: C:\B\AI\_TEAI\TEAIVideoMaker

📁 Loading classes from AllClasses.ps1

✅ All classes loaded.

✅ Class Load Check: ✅ Img .. ✅ Aud .. ✅ Sect .. ✅ TPRec .. ✅ Inst .. ✅ Prj .. ✅ Clt .. ✅ iCampaign ..

✅ Classes loaded. Continuing...

🎬 25-11-25 17:40:40 Running campaign from C:\B\AI\_TEAI\TEAIVideoMaker\TEAIVideoMaker.ps1 at 25-11-25 17:40:40

CONFIG: CampaignId: TEAIChatbotLI,

UseSavedTPts=True, OpenBrowserImgTabs=False

UseSavedTPRecs=False, Mode=Prod

font\_size=76, Cap\_m=70, Cap\_n=40, MaxCapLen=92

Show-Asset-Files: ============================== Showing Asset Files for CampaignId: TEAIChatbotLI

srcFolder: C:\B\AI\_TEAI\TEAIVideoMaker\Modules\IO\..\..\Assets\TEAIChatbotLI\Scripts

mp3Folder: C:\B\AI\_TEAI\TEAIVideoMaker\Modules\IO\..\..\Assets\TEAIChatbotLI\Audio

jpgFolder: C:\B\AI\_TEAI\TEAIVideoMaker\Modules\IO\..\..\Assets\TEAIChatbotLI\Image

1

Show-Asset-Files: 1 ass files:

Assets\TEAIChatbotLI\Scripts\VidCfg.ass

2

Show-Asset-Files: 3 List.txts:

Assets\TEAIChatbotLI\Scripts\AudList.txt

Assets\TEAIChatbotLI\Scripts\concat\_list.txt

Assets\TEAIChatbotLI\Scripts\ImgList.txt

3

Show-Asset-Files: 0 mp3s:

3

Show-Asset-Files: 0 jpgs:

5

📦 Build-TPRecs: Campaign: 'TEAIChatbotLI'

📦 Build-TPRecs: TPRecs: Generate TPRecs from SAVED TPts for Campaign

📦 Get-TPRecs: Using SAVED TPts for Campaign: 'TEAIChatbotLI'

📦 Get-TPRecs: Get SAVED TPts json from ..\..\AA\_Data\SavedTPts\SavedTPts.json

🌐 Get-TPRecs: TPts: 8 loaded for 'TEAIChatbotLI'

✅ Get-TPRecs: TPRecs: Built 8

Build-TPRecs: Create Captions and Keywords

Build-TPRecs: Some TPRecs missing Kws. Calling OpenAI to generate Caps and Kws.

Get-OpenAI-TPts: Each RestMethod response logged to ..\..\AA\_Data\RESTMethod\_Responses\TPtsAfterPrmts\_RawResponse\_<TPtId>.json

Get-OpenAI-TPts: Processing [8] TPts via OpenAI with AI\_Prompts\PromptTPtToCapKws.ps1

Loading | 8

🪙 Get-OpenAI-TPts: Token usage: 3954 [1:514][2:483][3:492][4:489][5:491][6:494][7:486][8:505]

TEAIVideoMaker.ps1: Two Pexels pages opened per TPt: Search by Kws & TPt

TEAIVideoMaker.ps1: Select 1 image TPt. D/L it. Rename to 'TPt\_<Id>\_Img.jpg'. Save to Assets\TEAIChatbotLI\Image

Get-Image-Asset-Files: NO IMAGE FILES FOUND IN: [].

TEAIVideoMaker.ps1: No image files found in . Creating images via Pexels.

Show-TPRecRaw: Show TPts in TPRecs:

1 🟦 TPt: Introducing TEAIChatbot - a distributed, serverless, event-driven platform for multi-channel conversion. -------------------------------- -------------------------------- --------------------------------

2 🟦 TPt: User comments on social media trigger the chatbot via a branded CNAME endpoint.

3 🟦 TPt: The CNAME routes requests into an Azure Function for normalization, auth, and flow dispatch.

4 🟦 TPt: The flow engine executes steps, manages state, and replies via the original channel.

5 🟦 TPt: If no deterministic match, the system falls back to OpenAI for dynamic response generation.

6 🟦 TPt: The user is guided to a landing page with product links, checkout, and payment options.

7 🟦 TPt: Landing pages are structured for conversion - headline, proof, CTA, and payment integration.

8 🟦 TPt: Download the full 52-page whitepaper to see how TEAIChatbot delivers conversion across channels.

Show-TPRecRaw: Show Captions:

1 🟨 Cap:(98)⚠️ Unleash \N the future of interaction \N with TEAIChatbot. \N A seamless, \N multi-channel journey.

2 🟨 Cap:(56) Comments ignite a chatbot \N through a branded pathway!

3 🟨 Cap:(74) CNAME guides the flow \N into Azure Functions \N for clarity and security.

4 🟨 Cap:(89) A digital conduit of tasks \N Balancing moments and messages.\N Flowing through channels.

5 🟨 Cap:(70) When certainty fades, \N OpenAI steps in.\N Dynamic responses await.

6 🟨 Cap:(78) Explore a world of choices \N at your fingertips. \N Seamless payments await.

7 🟨 Cap:(72) Crafting landing pages \N for soaring conversions \N with core elements.

8 🟨 Cap:(84) Unlock insights with our \N 52-page whitepaper!\N Discover the power of TEAIChatbot.

Show-TPRecRaw: Show Keywords:

1 🟧 Kws:( 1) multi-channel, serverless, event-driven, technology, innovation, digital, conversion, chatbot, Azure Functions, Platform

2 🟧 Kws:( 1) social, comments, engagement, automation, interaction, chatbot, Azure Functions, Platform

3 🟧 Kws:( 1) routing, normalization, authentication, dispatch, cloud computing, chatbot, Azure Functions, Platform

4 🟧 Kws:( 1) tasks, state, communication, chatbot, Azure Functions, Platform

5 🟧 Kws:( 1) response, intelligence, fallback, chatbot, Azure Functions, Platform

6 🟧 Kws:( 1) landing page, shopping, checkout, payment, user experience, chatbot, Azure Functions, Platform

7 🟧 Kws:( 1) headline, cta, integration, proof, chatbot, Azure Functions, Platform

8 🟧 Kws:( 1) whitepaper, channels, insights, business, chatbot, Azure Functions, Platform

Show-TPRecRaw: Show Image Pexels URLs:

8 🖼️ https://www.pexels.com/search/Introducing%20TEAIChatbot%20-%20a%20distributed%2C%20serverless%2C%20event-driven%20platform%20for%20mullti-channel%20conversion.%2F

8 🖼️ https://www.pexels.com/search/User%20comments%20on%20social%20media%20trigger%20the%20chatbot%20via%20a%20branded%20CNAME%20endpoint.%%2F

8 🖼️ https://www.pexels.com/search/The%20CNAME%20routes%20requests%20into%20an%20Azure%20Function%20for%20normalization%2C%20auth%2C%20and%%20flow%20dispatch.%2F

8 🖼️ https://www.pexels.com/search/The%20flow%20engine%20executes%20steps%2C%20manages%20state%2C%20and%20replies%20via%20the%20original%200channel.%2F

8 🖼️ https://www.pexels.com/search/If%20no%20deterministic%20match%2C%20the%20system%20falls%20back%20to%20OpenAI%20for%20dynamic%20responsse%20generation.%2F

8 🖼️ https://www.pexels.com/search/The%20user%20is%20guided%20to%20a%20landing%20page%20with%20product%20links%2C%20checkout%2C%20and%20payyment%20options.%2F

8 🖼️ https://www.pexels.com/search/Landing%20pages%20are%20structured%20for%20conversion%20-%20headline%2C%20proof%2C%20CTA%2C%20and%20paymment%20integration.%2F

8 🖼️ https://www.pexels.com/search/Download%20the%20full%2052-page%20whitepaper%20to%20see%20how%20TEAIChatbot%20delivers%20conversion%20accross%20channels.%2F

Show-TPRecRaw: OPENING IMAGE URLS IN BROWSER TABS...

1 🖼️ Simplified URL: https://www.pexels.com/search/Introducing TEAIChatbot - a distributed%2C serverless%2C event-driven platform for multii-channel conversion.%2F

2 🖼️ Simplified URL: https://www.pexels.com/search/User comments on social media trigger the chatbot via a branded CNAME endpoint.%2F

3 🖼️ Simplified URL: https://www.pexels.com/search/The CNAME routes requests into an Azure Function for normalization%2C auth%2C and flow ddispatch.%2F

4 🖼️ Simplified URL: https://www.pexels.com/search/The flow engine executes steps%2C manages state%2C and replies via the original channel..%2F

5 🖼️ Simplified URL: https://www.pexels.com/search/If no deterministic match%2C the system falls back to OpenAI for dynamic response generaation.%2F

6 🖼️ Simplified URL: https://www.pexels.com/search/The user is guided to a landing page with product links%2C checkout%2C and payment optioons.%2F

7 🖼️ Simplified URL: https://www.pexels.com/search/Landing pages are structured for conversion - headline%2C proof%2C CTA%2C and payment inntegration.%2F

8 🖼️ Simplified URL: https://www.pexels.com/search/Download the full 52-page whitepaper to see how TEAIChatbot delivers conversion across cchannels.%2F

Show-TPRecRaw: Audio TTS Files:

Show-TPRecRaw: 0 Audio TTS Files in C:\B\AI\_TEAI\TEAIVideoMaker\Modules\AA\_Incoming\_Text\_Html\_TPRecs\_Tps\..\..\Assets\TEAIChatbotLI\Audio

Get-Image-Asset-Files: NO IMAGE FILES FOUND IN: [].

Get-Audio-Asset-Files: Create Audio Files via OpenAI TTS

📝 Set-AudList\_txt: Generating AudList.txt for TEAIChatbotLI. 8 mp3 are required.

🎤 Set-AudList\_txt: Audio folder is Assets\TEAIChatbotLI\Audio

🎤 Set-AudList\_txt: TPt Id 1: Audio file is missing. --------------------------------

🎤 Set-AudList\_txt: TPt Id 2: Audio file is missing.

🎤 Set-AudList\_txt: TPt Id 3: Audio file is missing.

🎤 Set-AudList\_txt: TPt Id 4: Audio file is missing.

🎤 Set-AudList\_txt: TPt Id 5: Audio file is missing.

🎤 Set-AudList\_txt: TPt Id 6: Audio file is missing.

🎤 Set-AudList\_txt: TPt Id 7: Audio file is missing.

🎤 Set-AudList\_txt: TPt Id 8: Audio file is missing.

⚠️ Set-AudList\_txt: WARNING: Missing Aud files for TPt Ids: 1, 2, 3, 4, 5, 6, 7, 8. Getting .mp3s

⚠️ Set-AudList\_txt: TPt\_01\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio. --------------------------------

⚠️ Set-AudList\_txt: TPt\_02\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: TPt\_03\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: TPt\_04\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: TPt\_05\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: TPt\_06\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: TPt\_07\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: TPt\_08\_TTS.mp3 not found in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: WARNING: Missing Assets\TEAIChatbotLI\Audio .mp3s for TPt Ids: 1, 2, 3, 4, 5, 6, 7, 8.

🎤 Set-AudList\_txt: Found 0 TTS .mp3 files in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: WARNING: 8 TPts. Only 0 .mp3 files. Getting .mp3s

🎤 Set-AudList\_txt: Found 0 TTS .mp3 files in Assets\TEAIChatbotLI\Audio.

⚠️ Set-AudList\_txt: WARNING: 8 TPts. Only 0 .mp3 files. Getting .mp3s

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 1

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 2

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 3

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 4

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 5

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 6

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id

🎤 Set-AudList\_txt: Invoke-OpenAITTS for TPt Id 8

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_01\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_02\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_03\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_04\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_05\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_06\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_07\_TTS.mp3 is set.

🎤 Set-AudList\_txt: Assets\TEAIChatbotLI\Audio\TPt\_08\_TTS.mp3 is set.

🎤 Set-AudList\_txt: TPRec TPt TPt\_01\_TTS.mp3, Id 1 Aud.Dur = 7.031 --------------------------------

🎤 Set-AudList\_txt: TPRec TPt TPt\_02\_TTS.mp3, Id 2 Aud.Dur = 5.568

🎤 Set-AudList\_txt: TPRec TPt TPt\_03\_TTS.mp3, Id 3 Aud.Dur = 6.142

🎤 Set-AudList\_txt: TPRec TPt TPt\_04\_TTS.mp3, Id 4 Aud.Dur = 5.751

🎤 Set-AudList\_txt: TPRec TPt TPt\_05\_TTS.mp3, Id 5 Aud.Dur = 6.142

🎤 Set-AudList\_txt: TPRec TPt TPt\_06\_TTS.mp3, Id 6 Aud.Dur = 5.646

🎤 Set-AudList\_txt: TPRec TPt TPt\_07\_TTS.mp3, Id 7 Aud.Dur = 6.064

🎤 Set-AudList\_txt: TPRec TPt TPt\_08\_TTS.mp3, Id 8 Aud.Dur = 6.822

📝 Saved 8 entries to AudList.txt

Get-Audio-Asset-Files: All Audio files:

=======================================================================

⚠️ Set-OpenBrowserImgTabs: Fewer image files [ 0] than audio files [ 8]

Need to DL more images using opened browser tabs

Setting OpenBrowserImgTabs = true to open tabs for image selection

=======================================================================

💻 Opening browser tabs for image selection.

Show-TPRecRaw: Show TPts in TPRecs:

1 🟦 TPt: Introducing TEAIChatbot - a distributed, serverless, event-driven platform for multi-channel conversion. -------------------------------- -------------------------------- --------------------------------

2 🟦 TPt: User comments on social media trigger the chatbot via a branded CNAME endpoint.

3 🟦 TPt: The CNAME routes requests into an Azure Function for normalization, auth, and flow dispatch.

4 🟦 TPt: The flow engine executes steps, manages state, and replies via the original channel.

5 🟦 TPt: If no deterministic match, the system falls back to OpenAI for dynamic response generation.

6 🟦 TPt: The user is guided to a landing page with product links, checkout, and payment options.

7 🟦 TPt: Landing pages are structured for conversion - headline, proof, CTA, and payment integration.

8 🟦 TPt: Download the full 52-page whitepaper to see how TEAIChatbot delivers conversion across channels.

Show-TPRecRaw: Show Captions:

1 🟨 Cap:(98)⚠️ Unleash \N the future of interaction \N with TEAIChatbot. \N A seamless, \N multi-channel journey.

2 🟨 Cap:(56) Comments ignite a chatbot \N through a branded pathway!

3 🟨 Cap:(74) CNAME guides the flow \N into Azure Functions \N for clarity and security.

4 🟨 Cap:(89) A digital conduit of tasks \N Balancing moments and messages.\N Flowing through channels.

5 🟨 Cap:(70) When certainty fades, \N OpenAI steps in.\N Dynamic responses await.

6 🟨 Cap:(78) Explore a world of choices \N at your fingertips. \N Seamless payments await.

7 🟨 Cap:(72) Crafting landing pages \N for soaring conversions \N with core elements.

8 🟨 Cap:(84) Unlock insights with our \N 52-page whitepaper!\N Discover the power of TEAIChatbot.

Show-TPRecRaw: Show Keywords:

1 🟧 Kws:( 1) multi-channel, serverless, event-driven, technology, innovation, digital, conversion, chatbot, Azure Functions, Platform

2 🟧 Kws:( 1) social, comments, engagement, automation, interaction, chatbot, Azure Functions, Platform

3 🟧 Kws:( 1) routing, normalization, authentication, dispatch, cloud computing, chatbot, Azure Functions, Platform

4 🟧 Kws:( 1) tasks, state, communication, chatbot, Azure Functions, Platform

5 🟧 Kws:( 1) response, intelligence, fallback, chatbot, Azure Functions, Platform

6 🟧 Kws:( 1) landing page, shopping, checkout, payment, user experience, chatbot, Azure Functions, Platform

7 🟧 Kws:( 1) headline, cta, integration, proof, chatbot, Azure Functions, Platform

8 🟧 Kws:( 1) whitepaper, channels, insights, business, chatbot, Azure Functions, Platform

Show-TPRecRaw: Audio TTS Files:

Show-TPRecRaw: 8 Audio TTS Files in

C:\B\AI\_TEAI\TEAIVideoMaker\Modules\AA\_Incoming\_Text\_Html\_TPRecs\_Tps\..\..\Assets\TEAIChatbotLI\Audio

🎤 Image folder is Assets\TEAIChatbotLI\Image

📝 Saved 24 entries to ImgList.txt

📝 Set-AudList\_txt: Generating AudList.txt for TEAIChatbotLI. 8 mp3 are required. --------------------

🎤 Set-AudList\_txt: Audio folder is Assets\TEAIChatbotLI\Audio --------------------------------

🎤 Set-AudList\_txt: Found 8 TTS .mp3 files in Assets\TEAIChatbotLI\Audio.

🎤 Set-AudList\_txt: Found 8 TTS .mp3 files in Assets\TEAIChatbotLI\Audio.

🎤 Set-AudList\_txt: TPt Id 1 already has audio file assigned. Skipping TTS generation. -------

…

🎤 Set-AudList\_txt: TPt Id 8 already has audio file assigned. Skipping TTS generation.

🎤 Set-AudList\_txt: TPRec TPt TPt\_01\_TTS.mp3, Id 1 Aud.Dur = 7.031 --------------------------------

🎤 Set-AudList\_txt: TPRec TPt TPt\_02\_TTS.mp3, Id 2 Aud.Dur = 5.568

🎤 Set-AudList\_txt: TPRec TPt TPt\_03\_TTS.mp3, Id 3 Aud.Dur = 6.142

🎤 Set-AudList\_txt: TPRec TPt TPt\_04\_TTS.mp3, Id 4 Aud.Dur = 5.751

🎤 Set-AudList\_txt: TPRec TPt TPt\_05\_TTS.mp3, Id 5 Aud.Dur = 6.142

🎤 Set-AudList\_txt: TPRec TPt TPt\_06\_TTS.mp3, Id 6 Aud.Dur = 5.646

🎤 Set-AudList\_txt: TPRec TPt TPt\_07\_TTS.mp3, Id 7 Aud.Dur = 6.064

🎤 Set-AudList\_txt: TPRec TPt TPt\_08\_TTS.mp3, Id 8 Aud.Dur = 6.822

📝 Saved 8 entries to AudList.txt

Show-Asset-Files: ============================== Showing Asset Files for CampaignId: TEAIChatbotLI

srcFolder: C:\B\AI\_TEAI\TEAIVideoMaker\Modules\IO\..\..\Assets\TEAIChatbotLI\Scripts

mp3Folder: C:\B\AI\_TEAI\TEAIVideoMaker\Modules\IO\..\..\Assets\TEAIChatbotLI\Audio

jpgFolder: C:\B\AI\_TEAI\TEAIVideoMaker\Modules\IO\..\..\Assets\TEAIChatbotLI\Image

1

Show-Asset-Files: 1 ass files:

Assets\TEAIChatbotLI\Scripts\VidCfg.ass

2

Show-Asset-Files: 3 List.txts:

Assets\TEAIChatbotLI\Scripts\AudList.txt

Assets\TEAIChatbotLI\Scripts\concat\_list.txt

Assets\TEAIChatbotLI\Scripts\ImgList.txt

3

Show-Asset-Files: 8 mp3s:

Assets\TEAIChatbotLI\Audio\TPt\_01\_TTS.mp3 --------------------------------

…

Assets\TEAIChatbotLI\Audio\TPt\_08\_TTS.mp3

3

Show-Asset-Files: 0 jpgs:

5

💾 Set-ass-for-FFMPEG

file '..\Image/01Intro.png'

duration 7.031

file '..\Image/01Intro.png'

file '..\Image/02Entry.png'

duration 5.568

Assets\TEAIChatbotLI\Scripts\VidCfg.ass

2

Show-Asset-Files: 3 List.txts:

Assets\TEAIChatbotLI\Scripts\AudList.txt

Assets\TEAIChatbotLI\Scripts\concat\_list.txt

Assets\TEAIChatbotLI\Scripts\ImgList.txt

3

Show-Asset-Files: 8 mp3s:

Assets\TEAIChatbotLI\Audio\TPt\_01\_TTS.mp3 ---------------------------------

…

Assets\TEAIChatbotLI\Audio\TPt\_08\_TTS.mp3

3

Show-Asset-Files: 0 jpgs:

5

💾 Set-ass-for-FFMPEG

file '..\Image/01Intro.png'

duration 7.031

file '..\Image/01Intro.png'

file '..\Image/02Entry.png'

duration 5.568

Assets\TEAIChatbotLI\Scripts\AudList.txt

Assets\TEAIChatbotLI\Scripts\concat\_list.txt

Assets\TEAIChatbotLI\Scripts\ImgList.txt

3

Show-Asset-Files: 8 mp3s:

Assets\TEAIChatbotLI\Audio\TPt\_01\_TTS.mp3 --------------------------------

…

Assets\TEAIChatbotLI\Audio\TPt\_08\_TTS.mp3

3

Show-Asset-Files: 0 jpgs:

5

💾 Set-ass-for-FFMPEG

file '..\Image/01Intro.png'

duration 7.031

file '..\Image/01Intro.png'

file '..\Image/02Entry.png'

duration 5.568

5

💾 Set-ass-for-FFMPEG

file '..\Image/01Intro.png'

duration 7.031

file '..\Image/01Intro.png'

file '..\Image/02Entry.png'

duration 5.568

file '..\Image/02Entry.png'

duration 5.568

file '..\Image/02Entry.png'

file '..\Image/03Intent.png'

duration 6.142

file '..\Image/03Intent.png'

file '..\Image/04Sales.png'

duration 5.751

file '..\Image/04Sales.png'

file '..\Image/05Arch.png'

duration 6.142

file '..\Image/05Arch.png'

file '..\Image/06Loop.png'

duration 5.646

file '..\Image/06Loop.png'

file '..\Image/07Orch.png'

duration 6.064

file '..\Image/07Orch.png'

file '..\Image/08Outro.png'

duration 6.822

file '..\Image/08Outro.png'

💾 Set-ass-for-FFMPEG: Writing to VidCfg.ass

TEAIVideoMaker.ps1: TODO: PopulateImageLinks -TPRecs TPRec TPRec TPRec TPRec TPRec TPRec TPRec TPRec [Set-ass-for-FFMPEG sets images]

TEAIVideoMaker.ps1: TODO: FFMPEG: See Set-ass-for-FFMPEG and .\<CampaignId>.ps1

TEAIVideoMaker.ps1: TODO: Render loop?

✅ 25-11-25 17:41:35 TEAIVideoMaker finished in 55.8486206 sec. OpenAI tokens used: 3954

**SECTION** 3 — **Capability Overview**

(after you state “serverless distributed worker executes flows”).

**SECTION** 4 — System Layering Explanation, subsection 3.1 System Architecture

|  |  |
| --- | --- |
| **Overview — Layered Model Explanation.**  A layered decomposition of the platform.  Branded **CNAME** ingress  feeds Channels,  then Request Routing,  then Flow Execution that  consults Flow Definitions  plus Conversation State.  A parallel orchestration layer drives scale.  The Section 4 “layers, not boxes” explanation. img9 | **Figure 4-9 — Distributed Chatbot Platform Architecture,**  **Layered View** |

**SECTION** 5, subsection 4.1 Flow Definition Example

(right after introducing “event arrives, flow runs, reply returns”).

|  |  |
| --- | --- |
| A simplified “happy path” view of runtime behavior.  A user comment on a social platform triggers a webhook event, hits the CNAME ingress, then enters the Flow Engine, which returns a response back to the platform.  This is the minimal mental model to hold before diving into Flow Definitions.  Orientation: Narrow/vertical, R. Img4 | **Figure 5-2 — Social Comment**  **Trigger Path to Flow Response** |

**SECTION** 5 (early in Section 5 before deeper routing details).

|  |
| --- |
| **Figure 5- 3— Minimal** Ingress **Overview** |
| A sanity-check view of the distributed boundary.  Social users interact inside a platform, events route to your branded CNAME endpoint, then the Distributed Chatbot Platform handles everything beyond.  This is to help non-technical readers see the seam between “internet channels” vs our compute core. Img6 |

**SECTION** 5, subsection 4.4 Data Schema for Flow Definitions & User Context

(as the visual that motivates state + definition stores).

|  |
| --- |
| **Figure 5-4 — Runtime Flow Execution With State** |
| Clearest bridge between architecture vs schema. Img8 |

**SECTION** 5 — Network Flow Overview & Distributed System Architecture,

subsection 4.3

or right before 4.3 (sets context for Flow Engine + repo).

|  |
| --- |
| A diagram of a platform  AI-generated content may be incorrect.  **Figure 5-5 — Distributed Chatbot Platform Architecture** |
| This is the core platform topology.  A branded CNAME endpoint receives traffic from multiple channels, then routes into an Azure Function App hosting the Flow Engine.  The engine reads Flow Definitions, persists Conversation State, plus calls External APIs/Services.  This anchors the distributed model:  stateless compute for execution, separate stores for rules plus state. Img3 |

**SECTION** 5, subsection 4.2 or 4.3 as an alternate/cleaner version of Figure 8. (Figure 10)

|  |
| --- |
| **Figure 5-6 — Runtime Flow Execution With State (Alt Layout)** |
| Same concept as Figure 8 with a tighter visual rhythm.  It is a slide-like diagram for readers who skim.  It reinforces the interaction among CNAME ingress, Flow Engine, Flow Definitions, plus Flow State. 06Fallback |

**SECTION** 5 (top of Section 5, before 4.1).

|  |
| --- |
| **Figure 5-6 — High-Level Distributed Architecture Overview –** |
| The birds-eye view across channels, routing, compute, plus integrations.  Social platforms send requests into a CNAME Record that routes to the Chatbot Platform.  The platform drives Flow Engine execution, reads/writes State Store, plus calls External APIs.  This is a primer for the distributed flow details. imga |

**SECTION** 6 — Event-Driven Triggers & Automated Sales Flow (intro of Section 6).

|  |  |
| --- | --- |
| **Business-facing journey enabled by the platform**  A user comment in a social app becomes an event routed through your branded CNAME into the Flow Engine.  The deterministic steps  drive a reply back into the social channel,  then move the user to a  landing page, checkout, payment & confirmation.  Section 6 is a revenue automation layer,  not just a chatbot path. **img1** | **Figure 6-1 — User Journey**  **From Social Trigger to Purchase** |

**SECTION** 6 — Event-Driven Triggers & Automated Sales Flow (after Figure 1, as the full-stack view).

|  |  |
| --- | --- |
| **The Whole System on One Page**  **for Sales Flow Automation**  This end-to-end diagram ties the user journey to the platform internals.  Left lane shows the user experience in the social app through purchase.  Middle lane shows the technical backbone:  - platform event engine,  - Azure Function host  - Flow Engine  - state store  - outbound sender  Right lane lists real-world integrations (Telegram, Meta Graph, LinkedIn, Gumroad/Stripe). Img2 | A screenshot of a computer screen  AI-generated content may be incorrect.  **Figure 6-2 — End-to-End Tom English AI Sales & Chatbot Platform**  **Figure 6-1 — User Journey From Social Trigger to Purchase** |

**SECTION** 6 (mid-section, right after you explain product link tap).

|  |  |
| --- | --- |
| **“Comment-to-Customer” Automation**  **Platform-to-Payment**  A clean conversion path emphasizing  platform-to-payment continuity.  Social engagement  pushes users to Landing Page,  then Product card,  then Stripe/Gumroad checkout,  followed by confirmation.  imgi | A diagram of a sales conversion path  AI-generated content may be incorrect.  **Figure 6-20 — Sales Conversion Path From Social to Checkout** |

**SECTION** 6 (after we mention landing page + checkout mechanics).

|  |  |
| --- | --- |
| **View of the Shortest Possible Conversation**  **A compact complement to the richer sales diagrams**  A strict sales funnel slice.  After engagement, the user moves through  Landing Page → Product Page → Checkout → Payment.  imge | A diagram of a product  AI-generated content may be incorrect.  **Figure 6-15 — Checkout Funnel** |

**SECTION** 7 — Configuring api.tomenglishai.com as a Secure CNAME,

subsection **6.1** Platform → “send me events here”

or **6.2** Azure route → Function.

|  |
| --- |
| A screenshot of a computer  AI-generated content may be incorrect.  **Figure 7-13 — Serverless Backend Message Processing** |
| A practical routing pipeline.  Events:  - Enter via ingress,  - Pass through proxy validation,  - Enqueue for durability,  - Dequeue into stages: parse/transform/route/persist  Supports: Retries, backpressure, clean separation between public webhook surface vs internal execution. imgc |

**SECTION** 7, subsection **6.3** Outgoing Path (Replying)

or end of Section 7 as a “**why branded CNAME matters**” wrap-up.

|  |
| --- |
| A diagram of a chatbot  AI-generated content may be incorrect.  **Figure 7-14 — CNAME Ingress to Event Router** |
| Visualizing the branded domain as the stable inbound contract.  External webhook events:  Hit flow.example.com,  Resolve through CNAME to the function endpoint,  Enter a canonicalization stage  Event Router dispatch  This is the resilience story: stable DNS entry, flexible backend. imgd |

**SECTION 8 — Evolution of the Idea: Solving Engineering Questions Architecture Overview**

|  |  |
| --- | --- |
| **Flow Control & Conversation Logic**  **Figure:** Deterministic Flow vs AI Fallback (diamond: Step Match?)  (Already diagrammed earlier — no new figure) 06Fallback ?????? | **Figure 8-1 — Deterministic Step Logic with AI Fallback**  **The Flow Engine** |

|  |  |
| --- | --- |
| 06Fallback ?????? | **Figure 8-2 — OpenAI Fallback Execution Path**  **Visualizing the Flow** |

|  |  |
| --- | --- |
| **The OpenAI Fallback**  **--- Reuse the same fallback diagram**  **??? Figure:** *Visual Flow Designer Pipeline*  *(Editor → JSON → Publishing Platform)*  06Fallback ?????? | **Figure 8-3 — Visual Flow Designer Pipeline** |

**SECTION** **8 — Evolution of the Idea**, subsection 7.3 The Flow Engine (right after you describe step execution).

|  |
| --- |
| A diagram of a flowchart  AI-generated content may be incorrect.  **Figure 8-5 — Flow Engine Loop** |
| **The internal execution loop**  The engine:  loads a Flow Definition,  checks rules plus current state,  executes a Step,  generates a Reply,  then advances to the next Step until Finish.  The system remains deterministic, testable, plus replayable while still supporting branching logic. Img5 |

**SECTION** 8, subsection 7.4 The OpenAI Fallback.

|  |
| --- |
| **A diagram of steps and steps  AI-generated content may be incorrect.**  **Figure 8-7 — Deterministic Flow With OpenAI Fallback** |
| This figure captures the hybrid control model.  The Flow Engine first tries a deterministic Step match.  If a match exists, it runs the planned Action.  If no match exists, control passes to OpenAI to generate an Action plus Output.  This preserves reliability while still covering unplanned user input. Img7 |

**APPENDIX B**, subsection **B1** How chatbots show up in the world

(end of SECTION 2 Intro for earlier framing).

|  |  |
| --- | --- |
| This taxonomy shows where conversational systems live:  web, mobile, desktop, embedded,  plus progressive experiences.  Chatbots are not “a UI,” but a pattern that threads through multiple interface families,  often as an embedded conversational layer. imgf | **A diagram of a computer  AI-generated content may be incorrect.**  **Figure B-B1 — Interface Affinity Diagram** |

**SECTION** 11 — Operational Layer: Security, Logging & Cost Control, subsection 10.2 Logging or 10.3 Cost Monitoring.

|  |
| --- |
| **A diagram of a software structure  AI-generated content may be incorrect.**  **Figure 11-1 — Ops & Observability Layer** |
| Observability stack for a distributed chatbot platform.  Dashboards, alerting, query, visualization, plus log search feed centralized Logs, Metrics, Traces.  Those roll into real platform surfaces like log aggregation, time-series storage, plus distributed tracing.  It reinforces production readiness. imgh |

**SECTION 12** or **SECTION** 6 Closing Perspective if you want to stress go-to-market mechanics near the end.

|  |  |
| --- | --- |
| A practical layout recipe for high-conversion landing pages supporting our flows.  This makes the sales layer actionable. Imgj  Headline → Product Description → Proof → Image/Video → Call to Action. | **Figure 6-6 — Landing Page Structure** |

**APPENDIX C** — Future: Visual Flow Editor - No-Code for Business Users, Easy Maintenance for Developers.

|  |
| --- |
| A diagram of a flowchart  AI-generated content may be incorrect.  **Figure C-C1 — Visual Flow Designer Pipeline** |
| This is the future UX for authoring flows.  A Flow Designer uses a Visual Editor to produce structured JSON definitions, then publishes to the platform.  It ties back to the Flow Definition model while offering a productizable no-code layer. imgk |

**APPENDIX D** — Future Directions, subsection D1 Durable & Event-Driven Workflows.

|  |
| --- |
| A diagram of workflow integration  AI-generated content may be incorrect.  **Figure D-D1 — Durable Workflow Integration (Conceptual)** |
| This figure previews the next maturity step: **Durable orchestration wrapping the Flow Engine.**  Webhook traffic plus serverless functions feed Flow Engine Logic,  which invokes external function endpoints, then emits Final Output.  It signals future-proofing: long-running flows, retries, fan-out/fan-in patterns. imgg |

**APPENDIX D**, directly after **D-D1** as the concrete **Azure durable pattern**.

|  |  |
| --- | --- |
| **Canonical Durable Functions Flow**  Client calls HTTP Starter, Orchestrator coordinates Activity Functions, Task Hub persists checkpoints.  This shows the concrete hosting model you can plug beneath the Flow Engine to gain durability without rewriting business logic. imgg2 | **A diagram of workflow integration  AI-generated content may be incorrect.**  **Figure D-2— Durable Workflow Integration**  **(Azure Durable Functions Pattern)** |

..

..