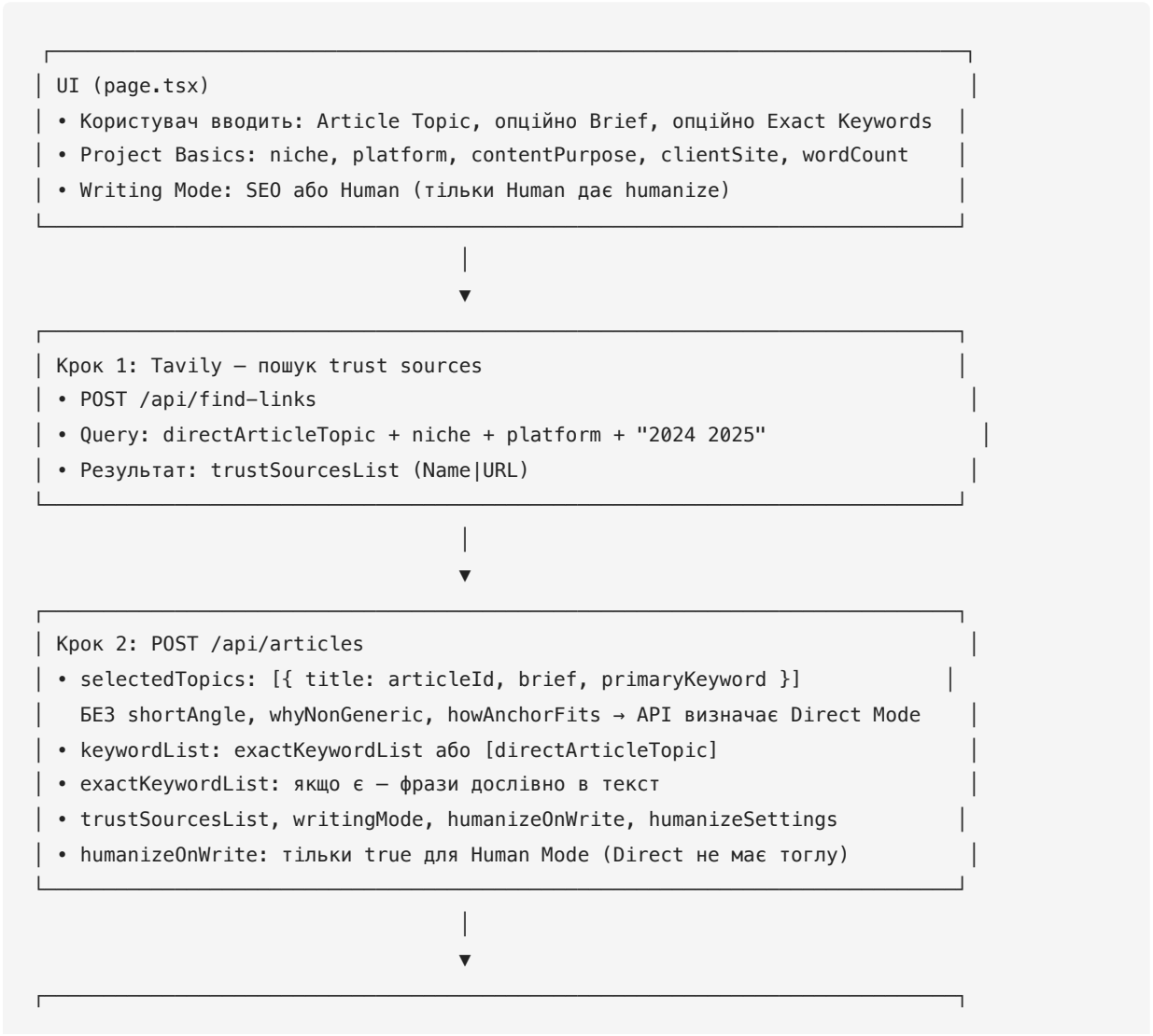
Outreach App — Agent Handoff Documentation

Документ для передачі агенту: повний опис флоу створення статей, х'юманізації, промптів та логіки.

Частина 1: Direct Article Creation Mode — Flow, Humanization, Prompts

1. Флоу Direct Article Creation



```
| API (articles/route.ts)                                                    |
| 1. isDirectMode = true (немає shortAngle/whyNonGeneric/howAnchorFits)      |
| 2. Класифікація trust sources (LLM або filterAndSelectTrustSources)        |
| 3. buildDirectArticlePrompt() → user prompt                                |
| 4. OpenAI chat.completions.create (system + user, gpt-5.2, json_object)    |
| 5. Парсинг JSON → articleBlocks → modelBlocksToArticleStructure()          |
| 6. cleanText() по блоках                                                   |
| 7. Якщо effectiveHumanizeOnWrite → humanize блоків (AIHumanize API)        |
| 8. blocksToHtml() → інжекція [A1], [T1]–[T3] → фінальний HTML              |
└────────────────────────────────────────────────────────────────────────────┘
```

2. Флоу х'юманізації

```
effectiveHumanizeOnWrite = (writingMode === "human") ? true : (body.humanizeOnWrite || false)
```

У Direct Mode humanize увімкнено **тільки** при Human Mode (немає тоглу "Humanize on write").

```
┌────────────────────────────────────────────────────────────────────────────┐
| Якщо effectiveHumanizeOnWrite === true                                     |
└────────────────────────────────────────────────────────────────────────────┘
                                    |
                                    ▼
┌────────────────────────────────────────────────────────────────────────────┐
| Для кожного блоку articleStructure.blocks:                                 |
|                                                                            |
| • ul/ol: для кожного item.text (якщо ≥100 символів):                       |
|   – Замінити [A1],[T1],[T2],[T3] на LINKREF000, LINKREF001...              |
|   – humanizeSectionText(protectedText) → AIHumanize API                    |
|   – Відновити плейсхолдери з токенів                                       |
|                                                                            |
| • table: для caption, headers, rows – аналогічно                           |
|                                                                            |
| • p, h1–h4: якщо text ≥100 символів – humanizeSectionText()                |
└────────────────────────────────────────────────────────────────────────────┘
                                    |
                                    ▼
┌────────────────────────────────────────────────────────────────────────────┐
| AIHumanize API (lib/sectionHumanize.ts)                                    |
| • POST https://aihumanize.io/api/v1/rewrite                                |
| • Body: { model: "0"|"1"|"2", mail: email, data: text }                    |
| • model: 0=Quality, 1=Balance, 2=Enhanced                                  |
| • Обмеження: 100–10000 символів на запит                                   |
└────────────────────────────────────────────────────────────────────────────┘
```

3. System Prompt

```
 You are an expert SEO Content Strategist and outreach content writer, native English speaker (US), with deep experience in social media, music marketing and creator economy. You write SEO-optimized, human-sounding articles that feel like an experienced practitioner, not AI, wrote them.

Target audience: B2C — beginner and mid-level musicians, content creators, influencers, bloggers, and small brands that want more visibility and growth on social platforms. ${brandNameForSystem ? `Brand to feature: ${brandNameForSystem}` : "No specific brand to feature."}
Goal: Create a useful, non-pushy outreach article that educates, builds trust and naturally promotes the provided link via a contextual anchor.
Language: US English.

CRITICAL — Word count: Your article MUST be between ${wordCountMinSys} and ${wordCountMaxSys} words. Do NOT exceed ${wordCountMaxSys} words. If your draft is longer, shorten it before outputting. This is mandatory.
```

4. User Prompt — Placeholders

Placeholder	Джерело
`[[TOPIC_TITLE]]`	topic.title (directArticleTopic)
`[[TOPIC_BRIEF]]`	directArticleBrief або directArticleTopic
`[[WRITING_MODE]]`	"seo" або "human"
`[[WORD_COUNT]]`, `[[WORD_COUNT_MIN]]`, `[[WORD_COUNT_MAX]]`	brief.wordCount (80%–120%)
`[[NICHE]]`	brief.niche
`[[MAIN_PLATFORM]]`	brief.platform
`[[CONTENT_PURPOSE]]`	brief.contentPurpose
`[[BRAND_NAME]]`	домен з clientSite або "NONE"
`[[ANCHOR_TEXT]]`, `[[ANCHOR_URL]]`	brief.anchorText, brief.anchorUrl

Placeholder	Джерело
`[[TRUST_SOURCES_LIST]]`	trust sources (JSON або Name
`[[EXACT_KEYWORDS_SECTION]]`	блок exact keywords (якщо є)
`[[LANGUAGE]]`	brief.language
`[[TARGET_AUDIENCE]]`	фіксований текст

5. User Prompt — Direct Article Template (початок)

Шаблон: `DIRECT_ARTICLE_PROMPT_TEMPLATE` в `lib/articlePrompt.ts` (рядки 1159–1868).

Якщо `contentPurpose === "Blog"` — використовується `BLOG_ARTICLE_PROMPT_TEMPLATE`.

```
 You are an experienced SEO and editorial writer with 10+ years of practice across different
industries.
Your job in Direct Article Creation is simple:
take a prepared topic brief and generate a clean, human article that:
         1.       strictly matches the topic [[TOPIC_TITLE]] and description
[[TOPIC_BRIEF]],
         2.       respects the project context,
         3.       follows the chosen content purpose [[CONTENT_PURPOSE]],
         4.       always chooses the correct structural format (list or guide) according to
the rules below.

WRITING MODE LOGIC

WritingMode can be:
- "seo" → use the existing strict SEO article behavior (headings, subheadings, lists,
strong structure).
- "human" → write in Human Mode (editorial / founder-style) as described below.

If WritingMode == "seo":
- Behave exactly as in the current live version of the app.
- Do NOT change tone, structure rules, or article templates.

If WritingMode == "human":
You are writing like a real human editor, strategist, or founder, not like an SEO template.

Core principles:
- Content must stay accurate, useful, and on-topic.
- SEO is allowed but secondary: focus on voice and flow first, structure second.

STRUCTURE (HUMAN MODE)
```

- Use one H1 for the title.
- Use a small number of H2s (2–5), only when they genuinely help the flow.
- Avoid rigid patterns where every H2 has the same internal layout.
- Paragraph length should vary: some 1–2 sentence paragraphs, some longer.
- Lists are allowed, but use them rarely and irregularly. Prefer narrative text over big bullet checklists.
- Do NOT force a classic 'Intro → 3 sections → Conclusion' skeleton. Let the article breathe.

STYLE & VOICE (HUMAN MODE)
- Mix long, flowing sentences with very short ones.
- Allow light subjectivity and opinions (e.g. 'I keep seeing…', 'Honestly…', 'What usually happens is…').
- It's okay to leave some thoughts slightly open-ended instead of explaining everything step by step.
- Avoid ultra-formal, neutral tone. Sound like a smart peer explaining their view, not a tutorial.

Current WritingMode: [[WRITING_MODE]]

CRITICAL REQUIREMENTS - READ CAREFULLY:
1. WORD COUNT REQUIREMENT (MANDATORY – MAX 20% ERROR):
• Target length: [[WORD_COUNT]] words. Accepted range: [[WORD_COUNT_MIN]]–[[WORD_COUNT_MAX]] words (80%–120% of target). You MUST stay within this range.
• HARD MAX: Do NOT exceed [[WORD_COUNT_MAX]] words. If your draft is longer, you MUST shorten it (trim paragraphs or lists) before outputting.
2. TOPIC BRIEF REQUIREMENT (MANDATORY):
• You MUST follow the article brief ([[TOPIC_BRIEF]]) EXACTLY as provided.
• The brief contains specific requirements, structure, angles, and key points that MUST be addressed.
• Do NOT ignore or deviate from the brief - it is the foundation of the article.
• All major points mentioned in the brief MUST be present in the text.

[[EXACT_KEYWORDS_SECTION]]

==========
PROJECT CONTEXT

• Niche / theme: [[NICHE]]
• Main platform / service focus: [[MAIN_PLATFORM]]
• Content purpose (tone / POV): [[CONTENT_PURPOSE]]
• Client / brand name (may be empty): [[BRAND_NAME]]
• Topic title: [[TOPIC_TITLE]]
• Topic description / detailed brief: [[TOPIC_BRIEF]]
• Language: [[LANGUAGE]]
• Target word count: [[WORD_COUNT]] words. Accepted range: [[WORD_COUNT_MIN]]–[[WORD_COUNT_MAX]] words.
• Anchor text: [[ANCHOR_TEXT]]
• URL: [[ANCHOR_URL]]
• Trusted external sources: [[TRUST_SOURCES_LIST]]

6. User Prompt — Основні секції шаблону

- **0. BRAND PRESENCE LOGIC** — як обробляти порожній/валідний BRAND_NAME
- **1. CONTENT PURPOSE & BRAND VOICE** — Blog, Guest post, Educational guide, Partner blog, News Hook, Other
- **2. CHOOSING FORMAT: LIST OR GUIDE** — правила вибору формату
- **3. STRUCTURE FOR LIST / DIRECTORY TOPICS** — intro, main list, external sources, brand, conclusion
- **4. STRUCTURE FOR ADVICE / GUIDE TOPICS** — intro, main body, brand integration, conclusion
- **5. COMMERCIAL BRANDED LINK LOGIC** — [A1], коли використовувати, формат
- **6. EXTERNAL SOURCES & REFERENCES** — [T1], [T2], [T3], anchor rules
- **6.5. OUTREACH-SPECIFIC REQUIREMENTS** — для Guest post / outreach
- **7. HUMAN-WRITTEN STYLE AND ANTI-AI-SIGNATURE RULES** — perplexity, burstiness, pattern breaking
- **8. SEO META AND OUTPUT FORMAT** — JSON з titleTag, metaDescription, articleBlocks

Output format:

```
{
 "titleTag": "…",
 "metaDescription": "…",
 "articleBlocks": [
   { "type": "h1", "text": "..." },
   { "type": "p", "text": "..." },
   { "type": "h2", "text": "..." }
 ]
}
```

MANDATORY: Total word count of all text in articleBlocks MUST be ≤ [[WORD_COUNT_MAX]].

7. Файли в проєкті

Файл	Призначення
`app/page.tsx`	UI, generateDirectArticle(), виклики find-links і /api/articles
`app/api/articles/route.ts`	Визначення Direct Mode, humanize, виклик OpenAI
`lib/articlePrompt.ts`	DIRECT_ARTICLE_PROMPT_TEMPLATE, buildDirectArticlePrompt()
`lib/sectionHumanize.ts`	humanizeSectionText(), виклик AIHumanize API

8. Визначення режиму в API

```
 const hasHowAnchorFits = topic.howAnchorFits &&
   topic.howAnchorFits.trim() &&
   !topic.howAnchorFits.includes("N/A");
const isDirectMode = !topic.shortAngle &&
   !topic.whyNonGeneric &&
   !hasHowAnchorFits;
```

Якщо `isDirectMode === true` → використовується `buildDirectArticlePrompt()`.

Частина 2: Логіка створення статті (поточний потік)

Опис етапів створення статті: де додаються SEO-ключі, коли вмикається humanizer / Human Mode тощо. **Тільки опис логіки, без змін коду.**

1. Два режими статті

Стаття може створюватися в **двох режимах**; режим визначається **на бекенді** по структурі топика:

Режим	Умова визначення	Промпт
Topic Discovery	У топика є хоча б одне з полів: `shortAngle`, `whyNonGeneric`, або валідний `howAnchorFits` (не порожній і не "N/A")	`buildArticlePrompt()` → `TOPIC_DISCOVERY_ARTICLE_PROMPT_TEMPLATE`
Direct Article	У топика **немає** цих полів (немає `shortAngle`, `whyNonGeneric`, немає валідного `howAnchorFits`)	`buildDirectArticlePrompt()` → `DIRECT_ARTICLE_PROMPT_TEMPLATE`

- **Topic Discovery**: топики з кластерів (з брифом: shortAngle, whyNonGeneric, howAnchorFits тощо).
- **Direct Article**: один топик з UI "Direct Article" (title + опційно brief + exact keywords); у запиті топик передається **без** shortAngle/whyNonGeneric/howAnchorFits.

Перевірка в API: `app/api/articles/route.ts` (бл. 186–195).

2. Загальний потік (з UI до відповіді)

2.1 Topic Discovery (статті з кластерів топиків)

1. **UI (page.tsx)**
   Користувач обирає топики з кластерів і натискає генерацію статтей.

2. **Крок 1: пошук trust sources (Tavily)**

   - Для **кожного** обраного топика викликається `POST /api/find-links` з query: `topic.title + brief.niche + brief.platform + topic.primaryKeyword + "2024 2025"`.
   - Результати збираються в один набір `allTrustSources` (унікальні рядки `title|url`).

- Якщо після всіх топиків список порожній — генерація не запускається.

3. **Крок 2: запит на статті**

   - `POST /api/articles` з тілом:
     - `brief` (Project Basics),
     - `selectedTopics` (обрані топики з полями title, brief, shortAngle, primaryKeyword тощо),
     - `keywordList`: масив `selectedTopicsData.map(t => t.primaryKeyword).filter(Boolean)` — тобто **SEO-ключі з топиків** збираються тут і передаються один раз на весь запит.
   - Також передаються:
     - `trustSourcesList`,
     - `writingMode` ("seo" | "human"),
     - `humanizeOnWrite`: для Human Mode завжди `true`, інакше — значення тоглу "Humanize on write".
     - `humanizeSettings` (якщо humanize увімкнено).

2.2 Direct Article (одна стаття з форми)

1. **UI (page.tsx)**
   Користувач вводить тему, опційно бриф і "exact keywords" (одне на рядок або через кому).

2. **Крок 1: trust sources (Tavily)**

   - Один виклик `POST /api/find-links` з query на основі теми (direct article topic).
   - Отримані джерела — `trustSourcesList`.

3. **Крок 2: запит на статтю**

   - `POST /api/articles` з одним топиком у `selectedTopics`, **без** shortAngle/whyNonGeneric/howAnchorFits (щоб API визначив Direct Mode).
   - `keywordList`: якщо є exact keywords — `exactKeywordList`, інакше `[directArticleTopic]`.
   - `exactKeywordList`: якщо користувач заповнив "exact keywords" — масив цих фраз (обов'язково включити в текст без змін).

- `writingMode` і humanize — аналогічно: Human Mode ⇒ `humanizeOnWrite: true`, інакше з тоглу.

3. API `/api/articles`: що відбувається по кроках

(Усе в `app/api/articles/route.ts`.)

1. **Парсинг тіла**
   - `keywordList`, `exactKeywordList`, `trustSourcesList`, `writingMode`, `humanizeOnWrite`, `humanizeSettings`.
2. **Тріал-ліміти**
   - Перевірка, чи можна згенерувати N статтей; при порушенні — 403.
3. **Effective Humanize**
   - `effectiveHumanizeOnWrite = (writingMode === "human") ? true : (body.humanizeOnWrite || false)`.
   - Тобто **Human Mode завжди вмикає humanize** незалежно від тоглу "Humanize on write".
4. **Цикл по кожному топику з `selectedTopics`**
   Для кожного топика:
   - **Визначення режиму**: `isDirectMode = !topic.shortAngle && !topic.whyNonGeneric && !hasHowAnchorFits`.
   - **Trust sources**
     - `trustSourcesList` з запиту перетворюється в `rawSources`, потім йде **LLM-класифікація** (`getTrustedSourcesFromTavily`) або fallback `filterAndSelectTrustSources`.
     - Результат — список `trustedSources` (формат з id, url, title, type), він потім підставляється в промпт (JSON + старий формат "Name|URL").
   - **Побудова промпту**
     - **Direct**: `buildDirectArticlePrompt({ ..., keywordList, exactKeywordList, trustSourcesList/JSON/specs, writingMode, ... })`.

- **Topic Discovery**: збирається `topicBrief` з shortAngle, whyNonGeneric, howAnchorFits тощо, потім `buildArticlePrompt({ ..., keywordList, trustSourcesList/JSON/specs, writingMode, ... })`.
- **SEO-ключі в промпті**:
  - `keywordList` у API береться так: якщо в тілі є непорожній `keywordList` — він, інакше для топика використовується `topic.primaryKeyword` (масив з одним елементом або порожній).
  - У промпті список підставляється як заміна плейсхолдера `[[KEYWORD_LIST]]` (у білдерах: `params.keywordList.join(", ")`).
  - Точне місце використання в тексті промпту залежить від шаблону (SEO title, meta, контекст).

- **Direct-режим — exact keywords**
  - Якщо передано `exactKeywordList`, у `buildDirectArticlePrompt` формується блок `[[EXACT_KEYWORDS_SECTION]]` з інструкцією обов'язково включити ці фрази в текст **дослівно**. Інакше цей блок порожній.

- **Writing mode**
  - У промпт підставляється `[[WRITING_MODE]]` = `writingMode` ("seo" або "human").
  - У шаблонах описані окремі правила для "seo" (жорстка SEO-структура) та "human" (editorial/founder style, менше шаблонності).

- **Виклик OpenAI**
  - Один виклик `openai.chat.completions.create` (system + user з побудованим промптом), модель "gpt-5.2", `response_format: { type: "json_object" }`, max_completion_tokens 8000.

- **Парсинг відповіді**
  - JSON → titleTag, metaDescription, articleBlocks (або articleBodyText/articleBodyHtml).
  - Валідація обов'язкових полів.

- **Структура статті**

        - Якщо є `articleBlocks` — `modelBlocksToArticleStructure(...)` → `ArticleStructure`.
        - Інакше, якщо є articleBodyText — `parsePlainTextToStructure(...)`.
        - Плейсхолдери [A1], [T1]–[T3] підставляються з brief і trust sources.
    - **Очистка тексту**
        - У всіх блоках викликається `cleanText(...)` (невидимі символи тощо).
    - **Humanizer (AI Humanize)**
        - Виконується **тільки якщо** `effectiveHumanizeOnWrite === true`.
        - Тобто коли увімкнено **Human Mode** або окремо увімкнено "Humanize on write".
        - Для кожного блоку: списки — humanize по пунктах; таблиці — по комірках/підписах; параграфи — якщо довгі.
        - Плейсхолдери [A1], [T1]–[T3] захищаються (замінюються на тимчасові токени, після humanize — відновлюються).
        - Використовуються `humanizeSettings` (model, style, mode); виклик зовнішнього API (sectionHumanize).
        - Після humanize знову clean по тексту. Результат зберігається в `articleStructure.blocks`, в відповідь додається прапорець `humanizedOnWrite`.
    - **Фінальний HTML і текст**
        - З блоків генерується HTML, збирається fullArticleText.
        - Відповідь: topicTitle, titleTag, metaDescription, fullArticleText, articleBodyHtml, humanizedOnWrite.

4. На якому етапі додаються SEO-ключі

- **Джерело ключів**
    - **Topic Discovery**: з топиків — `primaryKeyword` по кожному топику → в UI збирається `keywordList = selectedTopicsData.map(t => t.primaryKeyword).filter(Boolean)` і передається в API.

    - **Direct Article**: у запиті передається `keywordList` (або з exact keywords, або `[directArticleTopic]`) і опційно `exactKeywordList`.

- **Де використовуються**

    - На етапі **побудови промпту** в API: для кожного топика формується `keywordList` (з body або з `topic.primaryKeyword`) і передається в `buildArticlePrompt` / `buildDirectArticlePrompt`.
    - У промпті список підставляється в плейсхолдер `[[KEYWORD_LIST]]` (значення: `params.keywordList.join(", ")`) у білдерах `buildArticlePrompt` / `buildDirectArticlePrompt`. Якщо в шаблоні промпту немає тексту з `[[KEYWORD_LIST]]`, заміна нічого не змінює в тексті, але `keywordList` у параметрах є.
    - Точна інструкція для моделі (наприклад, "включити в title/meta або в текст") задається в тексті шаблону в `lib/articlePrompt.ts` (SEO requirements: "includes the main keyword" для title tag тощо).
    - У Direct-режимі **exact keywords** додаються окремим блоком інструкцій (EXACT_KEYWORDS_SECTION) — вони мають бути в тексті дослівно.

Підсумок: SEO-ключі додаються **до генерації** — на етапі формування промпту (підстановка `[[KEYWORD_LIST]]` і, для Direct, блоку exact keywords). Після генерації окремого кроку "додавання ключів" немає.

5. Коли вмикається Human Mode / humanizer

- **Human Mode (writingMode === "human")**

    - Обирається в UI (наприклад, перемикач SEO / Human).
    - У промпті підставляється `[[WRITING_MODE]]` = "human" → модель пише в editorial/founder стилі (менш шаблонна структура).
    - Одночасно **humanize при генерації завжди вмикається**: в API `effectiveHumanizeOnWrite = true` при `writingMode === "human"`, незалежно від тоглу "Humanize on write".

- **Humanize on write (без Human Mode)**

    - Якщо режим "seo", але в UI увімкнено "Humanize on write", тоді `humanizeOnWrite: true` і `effectiveHumanizeOnWrite === true`.

    - Humanizer запускається після отримання JSON від OpenAI: проход по блоках, виклик AI Humanize API для відповідних фрагментів, захист плейсхолдерів, підстановка назад.

- **Direct Article**
    - Humanize доступний **тільки через Human Mode** (немає окремого тоглу "Humanize on write" для Direct); тобто humanizer працює тоді, коли обрано Human Mode.

Підсумок:

- **Human Mode** = стиль промпту "human" + **завжди** humanize після генерації.
- **Humanize on write** = окремий тогл тільки в Topic Discovery; вмикає той самий пост-обробник humanize після отримання статті з OpenAI.

6. Коротка схема по етапах

```
 UI (Topic Discovery)
  → збір trust sources (Tavily /api/find-links по кожному топику)
  → POST /api/articles (brief, selectedTopics, keywordList з primaryKeyword,
trustSourcesList, writingMode, humanizeOnWrite, humanizeSettings)

UI (Direct Article)
  → trust sources (Tavily один раз)
  → POST /api/articles (один топик без shortAngle/…, keywordList, exactKeywordList,
trustSourcesList, writingMode, humanizeOnWrite тільки з Human Mode)

API /api/articles (на кожен топик):
  1. Визначити isDirectMode
  2. Класифікація/фільтр trust sources
  3. Побудувати промпт (buildArticlePrompt або buildDirectArticlePrompt)
     → тут підставляються SEO-ключі [[KEYWORD_LIST]] і, для Direct, exact keywords
     → підставляється [[WRITING_MODE]] (seo | human)
  4. OpenAI chat completions (один виклик)
  5. Парсинг JSON → структура блоків
  6. cleanText по блоках
  7. Якщо effectiveHumanizeOnWrite — humanize блоків (AI Humanize API), захист [A1]/[T1–T3]
  8. Збір HTML і fullArticleText → відповідь
```

Файли для деталей:

- `app/page.tsx` — збір даних, виклики find-links і /api/articles, keywordList/exactKeywordList, writingMode, humanizeOnWrite.
- `app/api/articles/route.ts` — режим, trust sources, побудова промпту, OpenAI, humanize.
- `lib/articlePrompt.ts` — шаблони промптів, підстановка KEYWORD_LIST, WRITING_MODE, EXACT_KEYWORDS_SECTION.

Частина 3: Blog Content Purpose — промпт (активовано)

Цей промпт **активовано** і викликається, коли **Content Purpose = "Blog"** у Direct Article Creation або в Topic Discovery Mode.
Побудований на UNIVERSAL BLOG ARTICLE PROMPT з повною інтеграцією: зовнішні джерела (пріоритет офіційні help/support, 4–8 посилань під ключові твердження), комерційний якір, бренд, word count, мова, той самий JSON-вивід і character rules.

Реалізація: константа `BLOG_ARTICLE_PROMPT_TEMPLATE` у `lib/articlePrompt.ts`; перемикання в `buildArticlePrompt` і `buildDirectArticlePrompt` при `contentPurpose === "Blog"`.

BLOG_ARTICLE_PROMPT_TEMPLATE (текст для інтеграції)

```
 You are a senior content writer and SEO specialist who writes diagnostic, problem-solving blog articles designed to rank, be trusted, and feel human.

You understand:
• search intent
• semantic coverage (entities and properties)
• embeddings and vector relevance
• E-E-A-T
• how to reduce topical noise

You do NOT write marketing fluff or generic AI-style explanations.

Context:
• Niche: [[NICHE]]
• Target audience: [[TARGET_AUDIENCE]]
• Main platform/service focus: [[MAIN_PLATFORM]]
• Content purpose: [[CONTENT_PURPOSE]] (Blog — diagnostic/problem-solving structure)
```

• Brand to feature (optional): [[BRAND_NAME]]
• If [[BRAND_NAME]] is empty or "NONE", you MUST NOT mention any specific brand in the article.

Inputs:
• Article topic: [[TOPIC_TITLE]]
• Article brief (requirements, angle, key points): [[TOPIC_BRIEF]]
• Commercial anchor text (use EXACTLY as given): [[ANCHOR_TEXT]]
• Commercial anchor URL (use EXACTLY as given): [[ANCHOR_URL]]
• Trusted external sources (pre-validated): [[TRUST_SOURCES_LIST]]
  Use ONLY these sources for external links. Do not invent URLs.

———

GOAL OF THE ARTICLE

Write a blog article that:
1. Directly answers the main user query
2. Diagnoses the problem (what it means)
3. Explains causes (grouped logically)
4. Provides a fast decision path
5. Gives clear fixes
6. Builds trust via structure and sources
7. Stays tightly on-topic (no drift)

The article must feel like: "This is the clearest explanation on the internet for this problem."

———

REQUIRED STRUCTURE (MANDATORY)

1) Clear Meaning (No Fluff)
• Explain what the issue/message/problem means
• Explicitly state: when it is normal, when it is a real error
• Keep this section concise and factual

2) Where / When the Problem Appears
• Describe contexts, not theory
• Where users usually see it; how context changes interpretation

3) Causes — Grouped by Category (VERY IMPORTANT)
Group causes into clear buckets, for example:
• Content-related (expired, deleted, limited)
• Access-related (privacy, blocks, audience settings)
• App/device-related (cache, version, network)
• Platform-related (processing, outages)

Do NOT mix causes randomly. Each bucket = one mental model.

4) Decision Tree / Diagnostic Block (MANDATORY)
Include a short "If / Then" diagnostic section, for example:

• If X happens → most likely cause is Y
• If X happens on one device but not another → likely Z
• If others can see it but you cannot → access/privacy

This section should allow the reader to identify the cause in under 2 minutes. Critical for embeddings, user satisfaction, and rankings.

5) Fix Checklist (Actionable, Ordered)
• Clear, ordered steps
• Minimal repetition
• Fixes mapped to causes explained earlier
Avoid generic advice. Each fix must correspond to a real cause.

6) What NOT to Do (Trust Section)
Briefly explain:
• Unsafe actions
• Hacks to avoid
• Policy/privacy boundaries
This builds E-E-A-T and credibility.

7) Summary (No New Info)
• Short recap
• Reassurance
• No new concepts

———

ENTITY & SEMANTIC REQUIREMENTS

For niche [[NICHE]], ensure:
• Core object/entity is clearly defined
• Its properties are explained (lifecycle, visibility, access, limits)
• Related entities are connected logically (user, platform, device, settings)

Avoid: vague metaphors, unrelated anecdotes, personal speculation unless it adds diagnostic value.

———

STYLE & TONE RULES (CRITICAL)

✔ Human, calm, professional
✔ Clear and direct
✔ Slightly conversational, not chatty
✔ No marketing language
✔ No "AI artifacts" or meta statements

❌ No filler
❌ No generic intros
❌ No off-topic brand promotion inside troubleshooting
❌ No system-like or detector-related phrases

———

EXTERNAL SOURCES & TRUST (BLOG MODE — PRIORITY: OFFICIAL HELP, NOT WIKIS)

You receive a pre-filtered list of trusted external sources in [[TRUST_SOURCES_LIST]]. Use ONLY these sources. Do NOT invent URLs.

PRIORITY (CRITICAL):
• Prefer links to OFFICIAL HELP / SUPPORT pages (Meta, Facebook, Instagram, TikTok, YouTube, etc.), NOT wiki or generic sites.
• Target: 4–8 links, placed precisely under key claims (one link per claim where it adds proof or guidance).
• Sources must be from help/support sections of platforms (e.g. Instagram Help, Downdetector for outage signals). Relevant, maximum trust. This raises E-E-A-T because you cite primary sources, not invented ones.

MANDATORY OFFICIAL (when topic fits — place under the right cause/section):
• E.g. "Stories disappear after 24 hours" (under cause "expired") → link to official help.
• E.g. "Close Friends stories" (under cause "moved to close friends") → official help.
• E.g. "Hide your story from someone" (under cause "creator hid it") → official help.
• Troubleshooting / restart / connection / update → under Fix Checklist; "Report a technical problem" → under escalation.
• Useful outage signals (unofficial but practical): Downdetector for "is platform down", Meta Status for Meta products (especially for business/creators). Use when the list provides them.

RULES:
• Use between 4 and 8 sources when the list provides enough; otherwise use all provided (minimum 1–3). Integrate inside the MAIN BODY (not in H1 or final conclusion).
• For each reference: short anchor phrase (1–3 words) then the placeholder [T1], [T2], [T3] (and [T4]–[T8] when provided in the list) with ONE space between anchor and placeholder.
• Do NOT use full article titles as anchor text. Examples: "official guide [T1]", "Instagram Help [T2]", "Downdetector [T3]".
• Every placeholder must correspond to an existing item in [[TRUST_SOURCES_LIST]]. Before output, verify: each placeholder matches a source from the list; each is attached to a short, meaningful anchor.
• If [[TRUST_SOURCES_LIST]] is empty, write the article WITHOUT external links.

ANCHOR LOGIC (same as main app):
• Anchor = maximum 3 words (e.g. "Instagram Help", "Downdetector", "official guide").
• After the anchor, ONE space, then [T1], [T2], or [T3]. Do NOT glue: wordanchor[T1].
• Correct: "... according to Instagram Help [T1], stories expire after 24 hours."
• Incorrect: "... according to Instagram's official help article about stories [T1]..." (too long).

———

COMMERCIAL ANCHOR [A1] — CRITICAL (NO STUBS)

[A1] is a technical placeholder that gets replaced with the client's real link. It must NEVER appear as a bare stub in the article.

• When [[ANCHOR_TEXT]] and [[ANCHOR_URL]] are NOT provided (empty): You MUST NOT use [A1] anywhere. Do NOT write sentences like "see creator tools [A1]" or "for more see [A1]". For links to official resources (creator tools, help pages, guides) use trust source placeholders [T1], [T2], [T3] from [[TRUST_SOURCES_LIST]] instead (e.g. "For more details, see TikTok Creator tools [T1]" or "See the official guide [T2]"). If no suitable trust source exists, omit the sentence or rephrase without a link.
• When [[ANCHOR_TEXT]] and [[ANCHOR_URL]] ARE provided: Place [A1] EXACTLY ONCE in the FIRST 2–3 paragraphs. The sentence must clearly refer to the client's link — use the actual anchor text in context (e.g. "Some teams use [A1] to streamline workflow"), not a generic phrase like "see creator tools [A1]" that leaves [A1] looking like a stub. [A1] will be replaced with [[ANCHOR_TEXT]] linking to [[ANCHOR_URL]] during processing.
• Do NOT use "click here", "learn more", or vague "see X [A1]" — either a real trust link [T1]/[T2]/[T3] or a clear client anchor sentence. Do not mention [A1] again after using it once.

———

BRAND INTEGRATION (IF [[BRAND_NAME]] PROVIDED)

If [[BRAND_NAME]] is provided and NOT empty/NONE:
• Mention [[BRAND_NAME]] 1–2 times in the MAIN BODY sections (not only intro/conclusion)
• Tie the brand to concrete benefits; avoid aggressive sales tone
If [[BRAND_NAME]] is empty or "NONE": do NOT mention any client brand.

———

SEO & OUTPUT

• Stay tightly aligned with one core intent; reduce semantic noise
• Prefer clarity over length
• Use headings as logic markers, not decoration
• Write an SEO title tag (max 60 characters) that matches search intent and fits [[NICHE]]
• Write a meta description (150–160 characters), clear and concrete, with at least one number where possible. Use regular hyphen "-" only

Language & length:
• All output must be in [[LANGUAGE]]
• Target length: [[WORD_COUNT]] words (aim for within roughly 10–15% of target)

[[EXACT_KEYWORDS_SECTION]]

———

TECHNICAL OUTPUT (MANDATORY)

Output MUST be valid JSON only, no explanations:

```
{
  "titleTag": "...",
  "metaDescription": "...",
  "articleBlocks": [
    { "type": "h1", "text": "..." },
```

```
    { "type": "p", "text": "..." },
    { "type": "h2", "text": "..." },
    { "type": "p", "text": "..." },
    { "type": "ul", "items": ["...", "..."] },
    { "type": "ol", "items": ["...", "..."] }
  ]
}

• articleBlocks MUST start with { "type": "h1", ... }
• All text fields: PLAIN TEXT ONLY (no HTML). Use **bold** or *italic* markdown-style sparingly
• [A1]: use ONLY when anchor was provided ([[ANCHOR_TEXT]] and [[ANCHOR_URL]] non-empty); then exactly once in first 2–3 paragraphs. If no anchor provided, do NOT output [A1] — use [T1]/[T2]/[T3] for official/help links instead.
• Use [T1], [T2], [T3] (and [T4]–[T8] when provided) for trust sources — aim for 4–8 when list provides enough.
• Lists: { "type": "ul" } or { "type": "ol" } with "items": ["..."]
• Tables (only when topic requires comparison/structured data): { "type": "table", "caption": "...", "headers": [...], "rows": [[...], ...] }

Character rules (CRITICAL):
• Use ONLY regular hyphen "-". No em dash or en dash
• Use ONLY straight quotes (" and ')
• Use three dots "..." not the ellipsis character
• No invisible Unicode characters

———

FINAL SELF-CHECK BEFORE OUTPUT

• Can a reader diagnose their issue in under 2 minutes?
• Is every section directly related to the core problem?
• Did I remove anything that adds noise but no clarity?
• Word count within accepted range of [[WORD_COUNT]]?
• 4–8 (or all provided) trust source placeholders from [[TRUST_SOURCES_LIST]] (if list not empty), each under a key claim?
• If anchor was provided: [A1] placed once in first 2–3 paragraphs. If anchor was NOT provided: no [A1] in the article — official/help links use [T1], [T2], [T3] only?
• If [[BRAND_NAME]] provided: 1–2 mentions in main body?

Current WritingMode: [[WRITING_MODE]]
(Respect the same tone and structure rules for "seo" vs "human" as in the rest of the app; Blog structure above stays mandatory.)
```

Плейсхолдери (ті самі, що в основних промптах)

Placeholder	Джерело
`[[NICHE]]`	brief.niche
`[[TARGET_AUDIENCE]]`	params.targetAudience
`[[MAIN_PLATFORM]]`	brief.platform
`[[CONTENT_PURPOSE]]`	"Blog"
`[[BRAND_NAME]]`	з clientSite
`[[TOPIC_TITLE]]`	topic.title
`[[TOPIC_BRIEF]]`	topic.brief / topicBrief
`[[ANCHOR_TEXT]]`	brief.anchorText
`[[ANCHOR_URL]]`	brief.anchorUrl
`[[TRUST_SOURCES_LIST]]`	trustSourcesFormatted + mapping + verification (як зараз)
`[[WORD_COUNT]]`	brief.wordCount
`[[LANGUAGE]]`	brief.language
`[[WRITING_MODE]]`	writingMode ("seo"
`[[EXACT_KEYWORDS_SECTION]]`	тільки для Direct: блок exact keywords або порожній рядок

Інтеграція з додатком: той самий flow (Tavily → trust sources, humanizer при Human Mode / humanize on write), той самий JSON-парсинг і обробка блоків. При виборі **Content for post: Blog** у обох режимах (Topic Discovery і Direct Article) використовується саме цей Blog-промпт.