Articles & Trivia – Product, API, RAG & UX Spec (v2)

0. Route Map / Surface Map

- **URL (main feed):** `https://connecwrk.com/articles-and-trivia/`
- **URL (logged-in detail view):** `https://connecwrk.com/articles-detail?articleId={id}` (inferred)
- **URL (public/offline detail view):** `https://connecwrk.com/article?articleId={id}` or deep-linked share URL
- **Share modal:** triggered from article detail / content share actions

Where this surface lives in the ecosystem

Left navigation (global sidebar on almost every screenshot): - MSME Connect - Freelancer Connect - Assignments - Talent Connect - MSME Jobs - **Articles and Trivia** ← this spec

Global top nav (header bar): - Search bar (placeholder changes contextually, e.g. “Search 'Freelancer'”, “Search 'MSME Job'”, etc.) - Home / Contacts / Messages / Notifications / Profile avatar

This matters for RAG because Connie should know how to route: - Any question like “Where can I read articles about marketing / business?” → point to Articles & Trivia. - Questions like “How do I share an article with my contacts?” → use Share Articles Popup flow.

---

1. Product Definition

1.1 What is Articles & Trivia?

Articles & Trivia is the platform’s content hub. It: - Shows a feed of editorial / curated content (cards with hero images) - Highlights trending or popular articles - Cross-promotes platform supply: featured Talent (Talent Connect), featured MSMEs (MSME Connect), and featured Freelancers (Freelancer Connect / Assignments) - Surfaces lightweight thought-leadership, tips, updates for MSMEs, freelancers, job seekers, and event talent

In short: this is both content marketing and discovery. It educates visitors, and also quietly advertises the people and businesses on ConnecWrk.

1.2 Business Goals

1. **Engagement / retention:** Give users a reason to scroll, learn, and come back even if they’re not currently hiring or applying.
2. **Demand generation:** Push visitors toward:
3. Hire a freelancer
4. Contact a speaker / trainer / artist (Talent Connect)
5. View a company (MSME Connect)
6. **Distribution:** Make articles easily shareable (WhatsApp, internal contacts) to drive referral traffic.

7. **SEO / Public Reach:** The public (logged-out) article detail view supports reading without full login, via an API-key protected “offline” endpoint.

1.3 Personas

Reader / Visitor (logged-in) - Lands on /articles-and-trivia/ - Scrolls cards - Opens full article detail - Might reshare to contacts or WhatsApp from Share modal - Might click a featured freelancer / MSME / talent and move into another vertical

Public Reader (logged-out) - Lands directly on an article link someone shared (e.g. WhatsApp) - Sees full read experience + related content and social links - Can’t access private contact info or internal messaging, but *can* keep browsing more public content

Talent / MSME / Freelancer being featured - Gets visibility via “Featured Talent Profiles”, “Featured MSMEs”, etc., rendered in the Articles & Trivia page sidebars and widgets - Traffic source for them

---

2. UI / UX Breakdown – Main Feed Page (`/articles-and-trivia/`)

Visually (per screenshot): masonry / grid layout of article cards with high-res images. Each tile shows: - Hero image (full-bleed image top) - Title (e.g. “Why ConnecWrk is a Game-Changer for MSMEs, Freelancers and Creators”) - Byline / source (e.g. “by ConnecWrk – 19 October 2025”) - Sometimes a short excerpt (business / finance / skills / industry topics)

2.1 Header Bar (global)

- Search input with search icon and submit chevron.
- Home, Contacts, Messages, Notifications, Profile avatar.
- Messages -> `https://connecwrk.com/messaging/`
- Contacts -> contact list view
- Profile avatar -> profile / settings options

2.2 Left Sidebar (global)

- User mini-card (avatar, name, profile completion %, connections count, profile views)
- CTA cards like:
- “MSMEs, Enhance Visibility — Create your business page →” (links into MSME Connect onboarding)
- “Hello, Freelancers — Create your profile here →” (links into Talent / Freelancer onboarding depending on context)
- Nav list:
- MSME Connect
- Freelancer Connect
- Assignments
- Talent Connect
- MSME Jobs
- Articles and Trivia (current)

Connie RAG note: this left sidebar is critical lore. If the user asks “Where do I create my business page?” Connie can answer: *From left sidebar > ‘MSMEs, Enhance Visibility — Create your business page’, which leads to MSME Connect onboarding.*

2.3 Main Content (center column)

- Grid of article cards (2–3 columns depending on viewport)
- Each card includes:
- Image
- Headline
- Metadata: author (usually “by ConnecWrk”), timestamp (“10 October 2025”), category context
- Some cards include microtopics, e.g. marketing, business policy, finance, “IRCTC ticketing, speed post…” etc.
- Cards include recent topical business/news style posts, plus evergreen career / freelancer content:
- "Six Weeks To Freelance Fame"
- "Build Your Personal Brand As A Freelancer"
- Macro/econ style cards like “Life insurers post 6% rise…”

2.4 Right-Hand / Secondary Sections

Depending on layout, right-hand column or lower sections can include: - **Advertisement block** - Labeled clearly as “Advertisement” - Static 250x250-ish creative - External link CTA (e.g. “Visit Now”) - **Featured Talent Profiles** / **Featured MSMEs** / **Featured Freelancers** widgets - Profile photo thumbnail, name/title - Short descriptor (e.g. “Branding and Marketing Speaker”, “Professional Singer”, “Web Development”) - Secondary tagline / summary - “View Profile” or implicit click target - Sometimes “View All →” CTA that leads into: - Talent Connect full list `https://connecwrk.com/talent-konectt/` / `https://connecwrk.com/more-talent-profiles/` - MSME Connect listing (MSME directory) - Freelancer listing `https://connecwrk.com/freelancers/`

2.5 Loading & Empty States

- On first render we show a loader/spinner while data is fetched.
- If an API call fails or returns no data:
- Main articles: show graceful empty fallback (“No articles found” + retry)
- Popular/trending: hide that section entirely
- Featured widgets: hide a specific widget, but keep rest of layout working (no page crash)

---

3. Article Detail Experience (Logged-in)

When a user clicks an article card from the feed, they land on a detail page that includes: - Full article title - Hero/featured image - Metadata (postedTimeAgo, total views, category, author “ConnecWrk Editorial”, estimated read time) - The full `articleDetails` HTML body - A sidebar of other/latest posts (thumbnails + small excerpt + timestamp) - A share entry point (Share popup)

3.1 Share CTA

- “Share” / “Share with Contacts” / “Send on WhatsApp”
- Opens Share Articles Popup (documented in §7)

3.2 Related / Latest Posts sidebar

- Renders additional articles via `/api/article/articlelist?page=1` (filtered to not repeat the current article)
- Shows: tiny image, title, postedTimeAgo
- Click → navigate to that article's detail page

3.3 View-count increment

- On viewing, frontend may call `POST /api/article/update-views` with `{ articleId }` to bump metrics.

---

4. Public / Offline Article Detail (Logged-out Reader Flow)

There is also a public/offline version of article detail that does **not** require a platform login, but *does* require a special API key header on the backend side. Purpose: allow viral content sharing and SEO.

4.1 Behavior / layout

- Similar to logged-in article detail: hero image, title, body, metadata.
- Includes a social row (icons linking to platform's official socials: Facebook, X/Twitter, Instagram, LinkedIn). This is part brand marketing, part trust.
- Sidebar still shows "Latest Posts" previews.
- No internal-only calls to action (like "Message this freelancer now") — keeps it safe for public.

4.2 Security model for offline view

- Uses endpoints that require an API key in headers, *not* Bearer auth:
- `GET /api/article/offlinearticledetail` (requires `x-api-key`) for a specific `articleId`
- `GET /api/article/offlinearticlelist` (requires `x-api-key`) for related article cards / "Latest Posts"
- That API key is controlled by backend, so even public pages still go through controlled access.
- Connie MUST NOT leak the API key.

---

5. API Reference

5.1 Get Articles List

Endpoint: `GET /api/article/articlelist` **Auth:** Bearer token (logged-in) **Params:** - `page` (number, required) — pagination index

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "article123",
      "articleTitle": "Understanding Digital Marketing Trends",
      "image": "digital-marketing.jpg",
      "postedTimeAgo": "2 days ago",
      "category": "Business",
      "readTime": "5 min read"
    }
  ],
  "totalPages": 5,
  "currentPage": 1
}
```

Errors: - 400 Bad Request (invalid `page`) - 401 Unauthorized (missing/invalid Bearer token) - 500 Server Error

Used for: - Main feed - Related / latest posts sections in detail views (authenticated)

5.2 Get Popular Articles

Endpoint: `GET /api/article/populararticle` **Auth:** Bearer token **Params:** - `page` (number, required — usually 1)

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "popular456",
      "articleTitle": "Top Business Strategies for 2024",
      "image": "business-strategies.jpg",
      "postedTimeAgo": "1 week ago",
      "viewCount": 1250,
      "engagementScore": 95
    }
  ]
}
```

Errors: 400 / 401 / 404 / 500

Used for: - “Popular / Trending” sections that highlight high-engagement content

---

5.3 Get Featured Talent (Talent Connect promo widget)

Endpoint: `GET /api/talent/featuredtalentlist` **Auth:** Bearer token

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "talent789",
      "name": "John Speaker",
      "profilePhoto": "speaker.jpg",
      "expertise": "Motivational Speaking",
      "rating": 4.8
    }
  ]
}
```

Errors: 401 / 404 / 500

Used for: - "Featured Talent Profiles" in the Articles & Trivia sidebar/footer - CTA to Talent Connect pages: - `https://connecwrk.com/talent-konectt/` - `https://connecwrk.com/more-talent-profiles/` - `https://connecwrk.com/talent-profile/` (create talent profile)

5.4 Get Featured MSMEs (MSME Connect promo widget)

Endpoint: `GET /api/msme/featuredmsmelisting` **Auth:** Bearer token

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "msme321",
      "businessName": "Tech Solutions Ltd",
      "industry": "Technology",
      "location": "Mumbai",
      "profileImage": "tech-solutions.jpg"
    }
  ]
}
```

Errors: 401 / 404 / 500

Used for: - “Featured MSMEs” block, which drives users toward MSME Connect / business pages

---

5.5 Get Featured Freelancers (Freelancer Connect promo widget)

Endpoint: `GET /api/project/featuredfreelancerlist` **Auth:** Bearer token

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "freelancer654",
      "firstName": "Sarah",
      "lastName": "Designer",
      "skills": ["UI/UX Design", "Figma", "Adobe Creative Suite"],
      "profilePhoto": "designer.jpg",
      "rating": 4.9
    }
  ]
}
```

Errors: 401 / 404 / 500

Used for: - “Featured Freelancers” widget - CTA to `https://connecwrk.com/freelancers/` and to Assignment / Hire flows

---

5.6 Get Article Details (Authenticated)

Endpoint: `GET /api/article/articledetail` **Auth:** Bearer token **Params:** - `articleId` (required)

Response:

```
{
  "success": true,
  "result": [
    {
      "id": "article123",
      "articleTitle": "Digital Marketing Strategies for 2024",
      "articleDetails": "<p>Complete article content in HTML format...</p>",
      "image": "digital-marketing.jpg",
      "postedTimeAgo": "3 days ago",
      "viewStatus": 1250,
      "category": "Business",
      "author": "ConnecWrk Editorial",
      "readTime": "5 min read",
      "tags": ["marketing", "business", "digital"]
    }
  ]
}
```

Errors: 400 / 401 / 404 / 500

Used for: - Logged-in article detail page content block

5.7 Update Article Views

Endpoint: `POST /api/article/update-views` **Auth:** Bearer token **Payload:**

```
{ "articleId": "article123" }
```

Response:

```
{
  "success": true,
  "message": "View count updated",
  "newViewCount": 1251
}
```

Used for: - Incrementing view counters when a logged-in user opens an article detail

---

5.8 Share Article (Tracked internal share)

Endpoint: `POST /api/article/share` **Auth:** Bearer token **Payload:**

```
{
  "articleId": "article123",
  "sharePlatform": "whatsapp",
  "shareMethod": "direct"
}
```

Response:

```
{
  "success": true,
  "message": "Article shared successfully",
  "data": {
    "shareId": "share_789",
    "shareUrl": "https://connecwrk.com/articles-detail?articleId=article123",
    "shareCount": 15
  }
}
```

Used for: - Analytics/logging when a user taps Share

5.9 Offline Article Detail (Public)

Endpoint: `GET /api/article/offlinearticledetail` **Auth:** REQUIRED API key header, not normal login. **Headers:** - `x-api-key: <server-provided key>` **Params:** - `articleId` (required)

Response: same shape as `articledetail`, minus any private data.

Errors: 400 / 401 (invalid key) / 404 / 500

Used for: - Public SEO / share landing page (non-logged-in visitor)

5.10 Offline Articles List (Public related posts)

Endpoint: `GET /api/article/offlinearticlelist` **Auth:** API key (`x-api-key`) **Params:** - `page` (required)

Response:

```json
{
  "success": true,
  "data": [
    {
      "id": "article456",
      "articleTitle": "Related Article Title",
      "image": "related-article.jpg",
      "postedTimeAgo": "1 week ago",
      "viewStatus": 800,
      "category": "Technology"
    }
  ],
  "totalPages": 10,
  "currentPage": 1
}
```

Used for: - "Latest Posts" / "Related Articles" sidebar on public article detail pages

---

5.11 Share with Contacts (Share Articles Popup)

Endpoint: `POST /api/share/sharewithmycontact` **Auth:** Bearer token **Payload:**

```
{
  "postId": "post123",
  "sharedata": "Check out this interesting article about digital marketing
trends!"
}
```

Response:

```
{
  "success": true,
  "message": "Post shared successfully with your contacts",
  "data": {
    "shareId": "share_789",
    "sharedAt": "2024-01-15T10:35:00Z",
    "recipientCount": 15
  }
}
```

Used for: - Pushing a post/article internally to your ConnecWrk contacts

---

5.12 Get Individual Post Details (for Share Preview)

Endpoint: `GET /api/share/individualsharepostdetail` **Auth:** Bearer token **Params:** - `postId` (required)

Response:

```
{
  "success": true,
  "data": [
    {
      "postId": "post123",
      "postTitle": "Digital Marketing Trends 2024",
      "postText": "Complete post content description...",
      "imageUrl": "https://example.com/image.jpg",
      "image": "post-image.jpg",
      "author": "John Doe",
      "createdAt": "2024-01-15T10:30:00Z"
    }
  ]
}
```

Used for: - Populating the Share Articles Popup preview so user can confirm what they’re sending

---

6. Share Articles Popup (UX + Validation)

6.1 Purpose

A modal that appears when the user clicks “Share”. It: - Previews the content to be shared (title, text, image) - Lets the user choose *how* to share: - Share with My Contacts (internal) - Send on WhatsApp (external) - Optionally lets them add a short personal message

6.2 User Flow

1. User clicks Share.
2. Frontend calls `/api/share/individualsharepostdetail?postId=...` to load preview.
3. Modal shows:
4. Post title (clickable to open original)
5. Image preview
6. Post text/excerpt
7. User selects share method:
8. **Share with My Contacts** → send payload to `/api/share/sharewithmycontact`
9. **Send on WhatsApp** → open WhatsApp deeplink with message pre-filled (includes article URL + optional custom message)
10. Show success toast + close (for internal share). For WhatsApp, modal can remain open.

6.3 Validation / Error Handling

- `postId` required
- Network/API failure → error toast + keep popup open so user can retry
- If user has zero contacts, backend may 404 / 404-like → show clear message
- Large custom message? we enforce length client-side to keep WhatsApp share neat

Connie safety rule:

> Connie can explain this flow (“Click Share → choose WhatsApp or Share with My Contacts”), but Connie should **not** fabricate personal emails / phone numbers of other users.

---

7. Error Handling (Global patterns)

7.1 Data Fetch Failures

- Show loading spinner while fetching.
- On failure:
- log error
- show friendly inline message ("Something went wrong, please try again.")
- Keep rest of page usable (partial render is okay).

7.2 Empty States

- Main feed: “No articles found.”
- Popular/trending: hide that widget.
- Featured Talent / MSME / Freelancer: hide specific widget and continue rendering the others.

7.3 Offline / Unauth Access

- If Bearer token is missing for private endpoints → 401 → user should be prompted to log in.
- If offline (public link) is accessed without `x-api-key` server-side, backend should 401. Frontend should fall back to a friendly “This content is unavailable” instead of crashing.

7.4 View Count / Share Count updates

- These calls (`/api/article/update-views`, `/api/article/share`, `/api/share/sharewithmycontact`) must gracefully no-op if they fail. Do **not** block article rendering.

---

8. Performance / Delivery / SEO

8.1 Performance tactics used or required

- Lazy-load images that are below the fold.
- Pre-size thumbnails / hero images to avoid layout shifts.
- Fetch in parallel:
- main article detail
- related posts
- featured talent/msme/freelancer
- Cache popular articles list briefly on client to avoid spamming `/populararticle`.
- Debounce pagination/scroll loads.

8.2 SEO / Public Reach

- Offline article detail uses public-friendly, indexable markup.
- Dynamic `<title>` and meta for the article title.
- Social icons (FB / X / Instagram / LinkedIn) encourage follow + credibility.
- Related Articles in sidebar improve internal linking.

---

9. Connie / RAG Integration Plan for "Articles & Trivia"

This section describes how this module’s knowledge will be ingested into the Connie assistant’s retrieval-augmented generation (RAG) backend.

9.1 Namespace strategy

- We maintain **one Pinecone index** (e.g. `connie-bot`).
- We ingest each logical product area into its own namespace.
- Articles & Trivia gets its own namespace:
- **namespace:** `articles_trivia`

Inside that namespace we attach `feature` metadata so retrieval can target the slice we care about: - `main_feed` (cards grid, layout, purpose of page) - `article_detail_auth` (logged-in article view behavior and `/api/article/articledetail`) - `article_detail_public` (public/offline view and `/offlinearticledetail`) - `share_popup` (share modal UX + /api/share/*) - `featured_widgets` (Featured Talent/MSME/Freelancer blocks + cross-surface navigation) - `api_spec` (all endpoint definitions) - `error_handling` (fallback patterns, offline rules) - `seo_performance` (performance, SEO and caching expectations)

This lets us: - Answer: “How does sharing work?” by retrieving only `share_popup` docs. - Answer: “Can someone without an account read an article?” by retrieving `article_detail_public`. - Answer: “Where do I see featured freelancers in Articles?” by retrieving `featured_widgets`.

9.2 Chunking strategy

We follow the global ingestion rules already defined for Connie: - Target ~800 word chunks (600–900 acceptable) - Overlap ~120 words so we don’t lose context on boundaries - Prefer splitting at strong headings (##, ###) - Each major heading section above (1., 2., 3., etc.) will likely become its own chunk or set of sub-chunks - Tables / lists of endpoints should stay intact in a single chunk so queries like “what’s the endpoint for popular articles?” retrieve that one cleanly

For API-heavy subsections: - We store each endpoint block (`Endpoint`, `Auth`, `Params`, `Response`, `Errors`, "Used for") as a single chunk with `content_type: "api_snippet"` - This makes dev-style questions (“what’s the payload for /api/article/share?”) retrieve that chunk directly, without noisy marketing copy.

9.3 Chunk metadata schema

Each upserted vector will include metadata like:

```
{
  "source": "articles_trivia_v2.docx",
  "namespace": "articles_trivia",
  "feature": "share_popup",
  "route": "/articles-and-trivia/",
  "content_type": "api_snippet" □ "ux_flow" □ "marketing_copy" □
"error_handling" □ "seo_performance",
  "doc_version": "v2",
  "updated_at": "2025-10-28",
  "chunk_index": 12,
  "hash": "<md5 of raw text>",
  "short_title": "Share Articles Popup",
  "api_path": "/api/share/sharewithmycontact"
}
```

This metadata enables: - **Routing** by intent: - If the user’s question includes verbs like “GET /api/…” or “what’s the payload”, the retriever will bias toward `content_type=api_snippet`. - If the question sounds like UX (“where do I click Share”), the retriever favors `content_type=ux_flow`. - **Debugging / provenance:** We can tell product, dev, compliance teams where Connie’s answer came from.

9.4 Deterministic chunk IDs + manifest for delta updates

Each chunk ID: `articles_trivia_pXX_cYY_<shortHash>` - `pXX`: pseudo-page / section number in this spec - `cYY`: chunk index within that section - `<shortHash>`: first 10 chars of md5 of chunk text

We keep a manifest file:

```
{
  "source": "articles_trivia_v2.docx",
  "doc_version": "v2",
  "chunks": [
    {"id": "articles_trivia_p01_c00_ab12cd34ef", "hash": "fullMd5Hash",
"chunk_index": 0},
    ...
  ]
}
```

Ingestion process: 1. Generate chunks + hashes. 2. Compare hashes to previous manifest. 3. Only upsert changed/new chunks to Pinecone. 4. Delete chunks that disappeared.

This keeps re-index cost low even if we revise copy or add new endpoints.

9.5 Guardrails / what Connie is allowed to say

Connie, when answering from this namespace: - ✅Can explain what Articles & Trivia is, how to open an article, how to share, where featured talent is shown. - ✅Can describe which API endpoints the UI calls and what parameters/payloads look like. - ✅Can mention that offline/public reading uses a secure API key header and that full internal contact details are not exposed publicly. - ❌Must **not** invent or expose phone numbers, emails, or private contact details for featured talent, freelancers, MSMEs unless the profile data is explicitly present in the ingested chunk. - ❌Must **not** leak any `x-api-key` values. It can say “requires x-api-key header” in general, which is in this spec. - ❌Must not claim user can DM talent directly from the public article page if that’s not possible; instead it should say you must log in / go to Talent Connect.

9.6 Regression Q&A set (post-ingestion validation)

We’ll test Connie with: 1. “What is Articles & Trivia on ConnecWrk and why should I care?” - Expect summary of §1 + cross-promotion aspect

1. “How do I share an article with my contacts?”
2. Expect Share Popup flow from §6 + `/api/share/sharewithmycontact` mention
3. “If I’m not logged in, can I still read an article someone sent me?”
4. Expect offline detail explanation from §4 and `offlinearticledetail` API
5. “Where do you pull ‘Featured Talent Profiles’ from on the Articles page?”
6. Expect reference to `/api/talent/featuredtalentlist` and Talent Connect
7. “What happens if the popular articles API fails?”
8. Expect fallback / hide section (Error Handling §7)

If Connie answers all of those correctly using only retrieved chunks from `articles_trivia` namespace, we consider the ingestion healthy.

---

10. Summary for Engineering / Product / Connie Team

- `/articles-and-trivia/` is not just blog. It's:
- Educational content
- Distribution surface for MSME Connect, Freelancer Connect, Talent Connect
- Engagement funnel for logged-in AND logged-out users
- The page is powered by modular APIs:
- Article list, popular list, article detail, share tracking, featured talent/MSME/freelancer widgets
- Public/offline versions gated by API key for SEO + virality
- We've defined:
- Complete UX flows (feed, detail, share modal)
- Error handling and fallback states
- Performance + SEO considerations
- Security boundaries (public vs logged-in)
- A full RAG ingestion plan (namespace, chunking, metadata, manifest, guardrails, regression Q&A)

This spec is considered `articles_trivia_v2`. Use it as the single source of truth for Connie ingestion under namespace `articles_trivia`. It is structured to scale, version, and update incrementally without reindexing the entire knowledge base.