

Premium Membership – Super Detailed Spec (v1)

1. Product Definition

Premium Membership is a paid (and in one case free-tier) upgrade layer for ConnecWrk users that unlocks enhanced visibility, contact access, promotional surfaces, and credibility signals across the platform.

The page at <https://connecwrk.com/premium-membership/> serves 3 jobs: 1. Explain what each plan includes (Basic / Basic Plus / Premium). 2. Let the user upgrade or purchase. 3. Enforce policy and payment rules (non-refundable, 1-year validity, ToS acceptance).

Why it exists

- Helps freelancers / MSMEs / talent get discovered faster.
- Gives ConnecWrk a recurring revenue stream.
- Gives buyers a “signal of trust” to decide whom to contact.

What happens when a plan is activated

- The user’s account gains entitlements (credits/month, profile boosts, featured posts, analytics, etc.).
- The system marks the plan as **active** with start date = purchase date and expiry = +1 year.
- UI across the platform (Freelancer Connect, Talent Connect, MSME Pages, Jobs, Assignments) should respect these entitlements.

NOTE: Membership fee, once paid, is **non-refundable** and validity is **1 year** from purchase. User must explicitly accept Terms & Conditions and Privacy Policy right before payment.

2. Primary URLs & Navigation Context

<https://connecwrk.com/premium-membership/>

- Public-facing (but requires login to actually initiate payment).
- Shows all plan cards and FAQs.
- Purchase buttons launch the payment/consent flow.

Terms acceptance modal

- Triggered when user clicks a payable plan CTA (e.g. "Subscribe", "Upgrade").
- Shows:
- Refund policy (non-refundable)
- Validity (1 year)
- Checkbox: [I agree to the above and accept ConnecWrk's Terms and Conditions and Privacy Policy](#)
- Primary action: [Proceed to Pay](#)
- Close / cancel action

Payment handoff

- After user accepts terms, we call backend to create the Cashfree order and open Cashfree's payment widget.
-

3. Personas & Value Prop

3.1 Freelancer / Talent / Individual Professional

- Wants inbound leads, visibility, discovery, trust.
- Pain today: Can't stand out from hundreds of profiles.
- Premium solves:
 - Higher search ranking
 - More featured placements
 - Contact credits to reach clients directly
 - Ability to showcase more work (more photos, social links)

3.2 MSME / Recruiter / Business Page Owner

- Wants to look credible to job seekers, freelancers, talent.
- Wants multiple business profiles under one login.
- Wants analytics on who looked at them.
- Premium solves:
 - Multiple company pages
 - Trust badges
 - Newsletter/features for brand exposure

3.3 ConnecWrk Platform / Admin

- Needs a consistent subscription source of truth.
 - Needs to know:
 - Which plan is active for a user
 - When it expires
 - Whether to unlock specific gated UI features
-

4. Plan Structure & Entitlements

We assume 3 tiers (names based on source text; final copy can vary slightly in UI):

4.1 Basic Plan (Free)

- **Credits / month:** 5 contact credits
- **Portfolio Photos:** up to 5
- **Business Pages:** up to 2 company / business profiles under the same user
- **Trust Badges:** 1 badge
- **Partner Benefits:** Access to baseline partner perks (e.g. SnackAmor)
- **Visibility:** Standard ranking in search
- **Featured Posts / Promo Surfaces:** None guaranteed

System notes: - This is the default for all new accounts. - Button state: usually "Current Plan" or no payment required.

4.2 Basic Plus Plan (₹1999 + GST / year)

- **Credits / month:** 15
- **Portfolio Photos:** up to 7
- **Featured Posts:** up to 4 per month (the profile / work can be pushed into featured carousels or spotlight modules)
- **Event / Webinar Access:** Invitation to certain events/webinars / networking programs
- **Search Boost:** Higher priority in internal search / listings
- **Partnership Benefits:** Expanded partner perks (discounts, collaborations, etc.)

System notes: - This tier is positioned as "I'm serious and want more leads now." - Only available if you don't already have an equal or higher plan. - If user has Premium, this button should be disabled with message like "You already have a higher plan".

4.3 Premium Plan (₹3999 + GST / year)

- **Credits / month:** 30
- **Portfolio Photos:** up to 12
- **Featured Posts:** up to 10 per month
- **Business Pages:** up to 3 MSME / company profiles attached to the same account
- **Trust Badges:** Multiple badges
- **Social Media Icons Integration:** Social links shown on public cards/profile modules
- **Profile Viewer Analytics:** See who viewed your profile / interest analytics
- **Newsletter Spotlight:** Appear in ConnecWrk outbound newsletters / marketing pushes
- **Search Boost:** Top-tier priority and promotion
- **Partnership Benefits:** Highest tier benefits, possibly exclusive deals / shoutouts

System notes: - Highest entitlement ceiling. - If user is already Premium, downgrade to cheaper plans must **not** be offered in UI.

4.4 Expiration / validity

- All paid plans: valid for **1 year from purchase date**.
- After expiry:
- User should gracefully fall back to Basic entitlements unless they renew.
- Analytics/history may remain, but premium-only visibility perks should be paused.

4.5 Refund policy

- **Strict:** "The membership fee, once paid, is non-refundable."
- Must be shown right before Proceed to Pay.

5. UI / UX Breakdown

5.1 Page Header / Banner

- High-level pitch of why Premium matters (credibility, visibility, access).

- Could include tagline like "Stand out. Get discovered faster."

5.2 Plan Cards Section

Each plan card shows: - Plan Name (Basic / Basic Plus / Premium) - Price + GST info (₹1999 + GST, ₹3999 + GST, etc.; Basic = Free) - Validity: 1 year - Bullet list of entitlements for that plan - CTA button at bottom

Button states must be dynamic: - **This plan is already active** (for the user's current plan) - **Upgrade** (for a higher plan) - **You already have a higher plan** (for any lower plan) - Disabled style if action is not allowed

5.3 Terms Modal (before payment)

Triggered by clicking a paid plan CTA.

Modal content: - Title: (ex) "Terms and Conditions" - Body copy: - "Thank you for considering the ConneCWrk Premium membership." - "The membership fee, once paid, is non-refundable and membership remains valid for 1 year from the date of purchase." - Explicit legal references to Terms and Conditions + Privacy Policy (linkable). - Checkbox: - ☐ I agree to the above and accept ConneCWrk's Terms and Conditions and Privacy Policy (must be checked to continue) - Primary CTA: - Secondary CTA:

5.4 FAQ / Help Section

- Expandable list (~8+ common questions):
 - What do I get in each plan?
 - How do I get featured?
 - How does contact credit work?
 - Will I be refunded if I cancel?
 - What happens after 1 year?
 - etc.

5.5 Post-Purchase State / Confirmation

- After successful payment:
- We should show a success confirmation / thank-you
- User's active plan should update in memory/state so the plans section re-renders (e.g. Basic Plus becomes "Active").

6. API Contracts & Data Flow

Below are the core APIs that power Premium Membership, plus how the frontend should call them.

6.1 Get Payment Plans

Endpoint:

Purpose: - Fetch list of available plans (planId, name, price, duration, features). - Drives the plan cards UI.

Auth: Bearer token (user must be logged in).

****Response (shape):**

```
{
  "success": true,
  "data": [
    {
      "planId": 1,
      "name": "Basic",
      "price": 0,
      "duration": "1 year",
      "features": ["Feature 1", "Feature 2"]
    },
    {
      "planId": 2,
      "name": "Basic Plus",
      "price": 1999,
      "duration": "1 year",
      "features": ["Feature 1", "Feature 2", "Feature 3"]
    }
  ]
}
```

Failure modes: - 401 Unauthorized → token invalid / expired - 500 Internal Server Error → show fallback "Plans unavailable" state

Frontend usage: - Map each plan to a card. - Use `planId` later when creating payment order.

6.2 Get User Profile (and current premium status)

Endpoint: `GET /api/auth/getuserprofile`

Purpose: - Check what the user already has (e.g. `{ planId: 2, planActive: true }`). - Used to disable "lower" plan CTAs.

Auth: Bearer token.

****Response (shape):**

```
{
  "success": true,
  "result": [
    {
      "premiumStatus": [
        {
          "planId": 1,
```

```

        "planActive": true
      }
    ]
  }
}

```

Frontend logic: 1. Determine `currentPlanId`. 2. For each plan: - if `plan.planId === currentPlanId` → button label: `This plan is already active` (disabled) - if `plan.planId < currentPlanId` → button label: `You already have a higher plan` (disabled) - if `plan.planId > currentPlanId` → button label: `Upgrade` (enabled)

This prevents downgrades.

6.3 Create Payment Order

Endpoint: `POST /api/payment/cashfreecreateorder`

Purpose: - Start the payment flow for a chosen plan. - Returns a Cashfree session/order that we hand off to the Cashfree SDK.

Auth: Bearer token.

Request body:

```

{
  "planId": 2
}

```

****Response (shape):**

```

{
  "success": true,
  "payment_session_id": "session_123456",
  "order_id": "order_789012"
}

```

Failure modes: - `400 Bad Request`: missing / invalid `planId` - `401 Unauthorized`: user not logged in - `422 Unprocessable Entity`: gateway rejected / plan not purchasable - `500 Internal Server Error`: payment infra down

Frontend flow: 1. User clicks "Upgrade" on a paid plan. 2. Terms modal opens. 3. User checks "I agree". 4. Frontend calls `cashfreecreateorder` with `planId`. 5. Use `payment_session_id` w/ Cashfree SDK to open payment sheet. 6. On success callback → show confirmation and re-fetch `getuserprofile`. 7. On failure → show toast and allow retry.

7. Payment / Compliance Flow

7.1 Preconditions

- User is logged in (Bearer token available).
- User selected a plan that is strictly higher than current plan.

7.2 Mandatory legal acknowledgment

Exact legal text that MUST appear in the modal before initiating payment:

"Thank you for considering the ConnecWrk Premium membership. The membership fee, once paid, is non-refundable and membership remains valid for 1 year from the date of purchase."

Plus:

"I agree to the above and accept ConnecWrk's Terms and Conditions and Privacy Policy."

Checkbox must be **checked** before enabling `Proceed to Pay`.

7.3 Cashfree session

- After acceptance, call `cashfreecreateorder`.
- Inject session ID into Cashfree checkout widget.
- Let Cashfree handle payment capture.

7.4 Post-payment state

- On success:
- Mark plan as active in backend.
- We expect backend to:
 - Store purchase date + expiry date (+1 year)
 - Record non-refundable policy acceptance
 - Write transaction details for audit/invoicing/compliance
- Frontend should:
- Refresh `getuserprofile`
- Update CTA states immediately (no manual reload required ideally)

8. Error Handling & Edge Cases

8.1 Plan load failure

- If `/api/payment/paymentPlanList` fails:
- Show a graceful fallback like: "Plans are temporarily unavailable. Please try again later."
- Keep page usable (FAQ can still render).

8.2 User profile load failure

- If `/api/auth/getuserprofile` fails:
- Assume no plan active (treat as Basic).
- Still allow upgrade.
- Log this for debugging; force re-auth if 401.

8.3 Payment order creation failure

- Show toast like: "We couldn't start the payment. Please try again."
- Keep modal open so user can retry without losing context.
- If 401, push to login.

8.4 Cashfree drop / cancel

- If user closes the Cashfree widget without paying:
- Do **not** activate plan.
- Do not show success.
- Allow retry from same plan CTA.

8.5 Downgrade attempt

- UI will already disable the button.
- Backend should ALSO enforce server-side: if user with Premium tries to create order for Basic Plus, respond `422` with a message like "You already have an active higher plan." This prevents forced downgrade via manual API hit.

8.6 Expired plan logic

- If plan expired:
 - Backend `getUserprofile` should either:
 - remove the `premiumStatus` entry, OR
 - set `planActive: false` and expose expiry.
 - Frontend should interpret that as "you are on Basic" and re-enable upgrade buttons.
-

9. Security, Compliance & Accounting

9.1 Auth & access

- All premium actions require a valid Bearer token.
- `/api/payment/cashfreecreateorder` MUST validate:
 - the user is authenticated
 - the `planId` is a valid upgrade path for that user

9.2 Audit trail

- Store:
 - `userId`
 - `planId` purchased
 - timestamp of purchase
 - expiry timestamp (+1 year)

- transaction/order ID from Cashfree
- explicit acknowledgment of Terms & Conditions

This protects ConnecWrk against refund disputes.

9.3 Non-refund enforcement

- The text we display in the modal is **part of legal proof**.
- We should capture `agreed=true` in the order request, or backend should log that the user saw/accepted it.

9.4 GST / tax

- UI should show `₹1999 + GST/year`, `₹3999 + GST/year` etc.
- The final payable amount surfaced by Cashfree must match what's displayed.

10. RAG / Connie Ingestion Requirements

Premium Membership needs to be fully answerable by Connie (the AI assistant). That means:

10.1 Namespace design

We will create a dedicated namespace inside the RAG index: - **Namespace:** `premium_membership`

This namespace will contain: 1. Plan definitions & entitlements 2. Pricing and GST statements 3. Refund & validity policy 4. Terms acceptance flow 5. Cashfree payment workflow 6. Upgrade rules (who can buy what) 7. FAQ content 8. Compliance notes (audit trail, non-refundable)

Why separate namespace: - It lets Connie answer "What do I get with Premium?" without pulling random info from Jobs / MSME Connect. - It also avoids hallucinating downgrade paths from other modules.

10.2 Chunking strategy for this namespace

Follow the global ingestion rules we defined earlier: - Target chunk size: ~600–900 words with ~100–150 word overlap. - Split on strong section boundaries (e.g. "4. Plan Structure & Entitlements"). - Each numbered section above becomes 1–2 chunks max. - The Terms & Conditions / refund statement **MUST** be its own chunk so Connie can quote/recall it precisely. - FAQ entries can be separate smaller chunks so Connie can retrieve very specific Q&A.

Special content types: - `content_type:"policy"` for refund/ToS/validity section. - `content_type:"plan_table"` or `content_type:"entitlements"` for benefits breakdown. - `content_type:"api_reference"` for endpoints.

10.3 Metadata per chunk

For every chunk inserted into Pinecone:

```
{
  "source": "premium_membership.docx",
  "namespace": "premium_membership",
  "chunk_index": 7,
  "doc_version": "v1.0",
  "content_type": "policy", // or entitlements / api_reference / faq
  "short_title": "Refund & Validity",
  "hash": "<md5-of-text>",
  "updated_at": "2025-10-28"
}
```

Why this matters: - `content_type` lets the retriever prioritize “policy” chunks for legal questions like “Can I get a refund?”. - `doc_version` + `updated_at` lets us rotate pricing in future without breaking history.

10.4 Retrieval guardrails for Connie

When the user asks Connie: - “Can I downgrade?” → Connie should retrieve upgrade/downgrade logic chunk and answer: “No, you cannot downgrade to a lower plan once you have a higher plan active.” - “Is it refundable?” → Connie must return exact policy: “No, membership fees are non-refundable. Valid for 1 year from purchase.” (This comes from `content_type: "policy"`). - “How do I pay?” → Connie should describe the Cashfree flow and the Terms checkbox BEFORE payment, not just “click pay”. (Good for compliance.) - “What do I get with Premium vs Basic Plus?” → Connie should pull entitlements from `content_type: "entitlements"` and generate a comparison.

10.5 API exposure in Connie answers

We want Connie to act as a support assistant, not a hacker. That means: - Connie *can* describe endpoints conceptually (“the system checks your plan using `/api/auth/getuserprofile`”) when talking to internal staff. - Connie *should not* instruct normal end users to call private backend APIs directly. - In production, we’ll likely have two modes: - **Internal / admin mode**: full API detail. - **Public / user mode**: no raw endpoint paths, just functional guidance.

We can implement this with role-based prompt prefixing.

10.6 Versioning / delta updates

When pricing, GST, or entitlements change: 1. Update `premium_membership.docx` (or equivalent source file). 2. Re-run ingestion for just that file. 3. Maintain deterministic chunk IDs (e.g. `premium_membership_c07_<hash>`), so only changed chunks get re-embedded and re-upserted. 4. Update `doc_version` and `updated_at` metadata.

This keeps Connie’s answers aligned with live pricing and legal text without re-indexing the entire knowledge base.

11. What Engineering Still Needs to Lock In

1. **Plan ID mapping** (are planId = 1/2/3 stable? We must not break existing users on renewal).
 2. **How expiry is stored** in `getUserprofile` (does backend return expiryDate? do we expose `planActive: false` once lapsed?).
 3. **Cashfree success callback contract** (what exact event do we listen for on frontend to refresh profile?).
 4. **Invoice / GST receipt storage** (if user asks support "Please send invoice", Connie should eventually be able to retrieve invoice path or instruct them how to request it).
 5. **Ability to surface badges / boosts** consistently across:
 6. Freelancer search cards
 7. Talent Connect cards
 8. MSME cards
 9. Job posters / recruiters
-

12. TL;DR for Stakeholders / Connie Cheat Answers

- Premium Membership = paid visibility + credibility + tools.
- All paid plans are annual. **No refunds.**
- You cannot downgrade once you're on a higher plan.
- Buttons on the Premium page reflect what you're allowed to buy.
- Before you pay, you MUST accept Terms & Conditions and acknowledge non-refund + 1-year validity.
- Payment runs through Cashfree using a session created by `/api/payment/cashfreecreateorder`.
- Connie will answer Premium questions using the `premium_membership` namespace in RAG, with chunks tagged as `entitlements`, `policy`, `api_reference`, and `faq` so it stays accurate and legally consistent.