Privacy & Data Controls – Super Detailed Spec (v1)

0. Glossary / Concepts

- **Privacy Policy page**: Public-facing legal disclosure of how ConnecWrk collects, uses, stores, and shares data.
- **Privacy Settings page**: Logged-in user's control panel for visibility, discoverability, contact permissions, etc.
- **Personal Data Popup**: Modal that lets the user edit sensitive fields like name, DOB, gender, and location.
- **Consent record**: Logged proof (versioned) that a user agreed to a specific Privacy Policy.
- **Grievance ticket**: An inbound complaint or request related to privacy / data rights.
- **RAG surface**: Connie AI assistant answers questions using retrieved (indexed) legal / privacy content instead of hallucinating.

All 3 surfaces work together: 1. Legal tells the user what happens ✅2. Settings lets the user control what happens 🔞3. Personal Data lets the user correct what's stored about them ♌4. Consent + grievance create legal audit trails 🟡5. RAG exposure makes it answerable in chat 🟫

---

1. Privacy Policy Page

URL (public): `/privacy-policy` (frontend)
Primary purpose: Communicate data practices, show effective dates, let users download / escalate concerns, and (if logged in) record consent.

1.1 Content responsibilities

- Inform users what data is collected (profile data, behavioral data, device/IP data, etc.)
- Inform users *why* (security, analytics, personalization, matching, messaging, etc.)
- Inform users *who can see what* (contacts, recruiters, public, search index, etc.)
- Provide legal basis + compliance references (IT Act 2000, SPDI Rules 2011, Digital Personal Data Protection Act 2023, etc.)
- Expose grievance officer contact path (escalation channel)
- Expose latest version + effective date
- Allow PDF download for legal records
- Power downstream consent logging (for audit readiness)

1.2 Backend contracts

Get Privacy Policy Content

`GET /api/legal/privacy-policy` - **Auth:** not required (public)
- **Goal:** front end renders latest policy text; Connie AI indexes this for RAG. - **Response:** - `version` (string, e.g. "2.1") - `effectiveDate`, `lastUpdated` - `content` (HTML string for rich display) - `language` (ex: `en`) - `regulatoryCompliance` (array of strings e.g. `IT Act 2000`, `SPDI Rules`

`2011`, `DPDPA 2023`) - **Fail states:** `500` → show fallback hardcoded text in UI, log error. RAG must keep previous successful version.

Download Privacy Policy (PDF)

`GET /api/legal/privacy-policy/pdf` - **Auth:** optional
- **Params:** `language` (optional, default `en`)
- **Returns:** PDF stream (Content-Disposition: attachment) - **Fail states:** `500` → show toast "Download temporarily unavailable".

Record Consent

`POST /api/user/privacy-consent` - **Auth:** required (Bearer token) - **When called:** - User signs up / first login after policy change - User explicitly taps "I Agree" CTA below policy - **Body:** - `policyVersion` (string) - `consentType` (must be `"privacy_policy"`) - `accepted` (bool) - `ipAddress` (string) - `userAgent` (string) - **Response:** `{ success, message, consentId }` - **Fail states:** - `400` missing fields → block submission, highlight required - `401` → force re-auth - `422` invalid version → re-fetch latest `/api/legal/privacy-policy` and retry with new version

Get Consent History

`GET /api/user/consent-history` - **Auth:** required - **Use cases:** - Show "You accepted v2.1 on 15 Jan 2025 from IP 103.x.x.x" in Settings > Privacy proof-of-consent - Provide evidence for compliance / audits - **Response Array items:** `consentId`, `policyVersion`, `accepted` (bool), `acceptedAt` (ISO timestamp), `ipAddress` - **Fail states:** - `401` → re-auth - `404` → "No recorded consent yet"

Submit Privacy Grievance / Escalation

`POST /api/support/grievance` - **Auth:** optional (guest can complain) - **Body:** - `name` (2–100 chars) - `email` (valid email) - `subject` (5–200 chars) - `message` (10–2000 chars, plain text) - `category` (required enum: `privacy_policy`, `data_access`, `data_deletion`, etc.) - `userId` (optional, if logged in) - **Response:** - `ticketId` (e.g. `GRV-2025-001`) - `estimatedResponseTime` (string like `"48 hours"`) - **Fail states:*** - `400` → client highlight specific invalid fields - `429` → "Too many submissions, please wait" - `500` → show generic "Could not submit right now"

1.3 Page layout / UX

1. **Header block**
2. Title: "Privacy Policy"
3. Version badge (`v2.1`)
4. Effective date / Last updated
5. **Policy body** (HTML from API)
6. Sections like "Data We Collect", "How We Use Data", "How Long We Keep Data", "User Rights", "Security", "International Transfers", etc.
7. **Compliance footer**
8. Mentions: IT Act 2000, SPDI Rules 2011, DPDPA 2023
9. Grievance Officer contact path → powered by `/api/support/grievance`
10. **Actions row (if logged in)**
11. Checkbox: "I agree to the current Privacy Policy"
12. CTA: "Accept & Continue"

13. On click → call `/api/user/privacy-consent`
14. **Download PDF link**
15. Calls `/api/legal/privacy-policy/pdf`

Graceful degradation: - If `/api/legal/privacy-policy` fails, we still render a static, bundled fallback text. We *must not* block signup/login on render failure.

1.4 RAG / Connie behavior for Privacy Policy

• We ingest:
• `/api/legal/privacy-policy` (HTML text, version, effectiveDate, compliance list)
• The FAQ-ish language around data usage, retention, user rights
• Escalation / grievance steps & SLA
• Retrieval intents Connie must handle:
• "What data do you store about me?"
• "How can I request deletion?"
• "Who do I contact about privacy?"
• "What laws do you follow?"
• "When did the policy last change?"
• Guardrails:
• Connie answers with sourced/grounded text only.
• Connie should **not** invent regulatory claims outside `regulatoryCompliance` list or retention promises that aren't in the policy.
• Connie can link user to grievance flow by summarizing `POST /api/support/grievance` ("We'll open a ticket; SLA ~48h").
• Version awareness:
• Connie should surface `version` + `effectiveDate` in answers so Legal can prove what user was told.

---

2. Privacy Settings Page (User-Controlled Visibility)

URL (authenticated): `/settings/privacy`
"Your privacy is in your hands" screen with two tab groups: Profile Settings + General Settings.

2.1 Goal

Give logged-in members granular control over: - Who sees what on their profile - How easily they can be discovered / contacted - Whether they want recruiters / mentions / indexing

Think of this as **policy application layer**. Privacy Policy says "we *can* do X". Privacy Settings says "please *do* or *don't* do X for me".

2.2 Backend contracts

Fetch Privacy Settings

`GET /api/settings/privacy` - Auth: required - Returns: - `profileSettings` - `contactsTabVisibility` (`all_members` | `contacts_only` | `none`) - `activityTabVisibility` (`all_members` | `contacts_only` | `none`) -

`searchEngineVisibility` (bool) - `contactListVisibility` (`all_members` | `connections` | `none`) - `generalSettings`
- `publicGroupsIndexing` (bool) - `openToOpportunities` (bool) - `allowMentions` (bool) - `lastUpdated` (ISO timestamp) - Fails: - `401` → force login - `404` → treat as defaults; display defaults in UI - `500` → show banner "Unable to load privacy settings"

Update Privacy Settings

`PUT /api/settings/privacy` - Auth: required - Body mirrors `GET` shape - Validation rules: - All required fields must be present, even if unchanged. - Enums must be recognized values. - Returns `{ success, message, data.updatedAt }` - Fails: - `400` → per-field validation error, highlight widget(s) - `401` → re-auth - `422` → invalid combination (e.g. contradictory choices) - `500` → show generic error toast, **do not** optimistically update UI state

Reset to Defaults

`POST /api/settings/privacy/reset` - Auth: required - Body: `{ "confirmReset": true }` - Returns new canonical defaults and timestamp. - Used by "Reset Privacy Settings" CTA.

Fetch User Profile (supporting data)

`GET /api/auth/getuserprofile` - Auth: required - Also returns `privacySettings` snapshot for hydration if `/api/settings/privacy` is not yet initialized.

2.3 UI structure / copy

Header block: - Shield / lock visual - Headline: "Your privacy is in your hands!" - Subcopy: "You decide who can view your activity, how you appear in search, and whether recruiters can reach out."

Tabs / sections: 1. **Profile Settings** - Contacts tab visibility (`all_members` vs `contacts_only` vs `none`) - Activity tab visibility - Allow my profile to appear in external search engines (toggle maps to `searchEngineVisibility`) - Who can see my contact list (`all_members`, `connections`, `none`) 2. **General Settings** - Allow my public groups / posts to be discoverable (`publicGroupsIndexing`) - Show me as "Open to Opportunities" to recruiters (`openToOpportunities`) - Allow people to @mention me (`allowMentions`)

Controls: - Inline "Edit" → turns rows into toggles / dropdowns - "Save" → calls `PUT /api/settings/privacy` - "Reset to Default" → calls `POST /api/settings/privacy/reset`

2.4 Error / edge behavior

- If `GET /api/settings/privacy` 500s:
- We still render skeleton rows with disabled controls
- Banner: "We're having trouble loading your privacy settings. Please try again later."
- If `PUT /api/settings/privacy` fails validation (400/422):
- Keep modal open
- Mark the specific field in red with descriptive helper text
- If `401` on any call:
- Immediate logout / re-auth flow to avoid phantom state

2.5 RAG / Connie behavior for Privacy Settings

We ingest: - Descriptions of each setting (what it means, who gets to see what) - Default values after reset - The fact that you *can* hide contacts/activity, limit mentions, opt out of recruiter visibility, etc. Connie must: - Explain what a toggle does in plain language ("If you set Activity Tab Visibility to `none`, no one can see your activity tab on your profile") - NOT flip settings itself (Assistant cannot claim "I've turned this off" — it can only guide user where to click) - Be privacy-positive: encourage cautious visibility for minors / sensitive users - Mention legal routes for account deletion / data removal → ties to Privacy Policy + grievance ticket

---

3. Personal Data Popup (Edit Personal Info Modal)

Surface: - Triggered from `/settings/personal-data`, `/personal-data`, or from "Edit" buttons next to Full Name, Gender, DOB, Location on profile/settings screens. - Appears as modal dialog titled "Edit Personal Data".

Example fields visible in screenshot: - Full Name (Lavanya) - Gender (Male / Female radio) - Date of Birth (Day / Month / Year dropdowns e.g. 22 / December / 2004) - Country (Select Country) - State (Select State) - City (free text / dropdown depending on region) - CTA: `Save` (or `Submit`) and `Cancel`

3.1 Business goals

- Let users self-correct legal identity / demographics (name, gender, DOB)
- Let users update location (country, state, city) which downstream affects:
- Search & match (freelance assignments near you, MSME geo, talent geo, recruiter filters)
- Tax / invoicing / KYC in future
- Platform analytics / trust signals
- Keep data legally accurate for consent age gates

3.2 Backend contracts

Get User Profile (prefill)

`GET /api/auth/getuserprofile` - Auth required - Returns: - `firstName`, `lastName` - `gender` - `dob` (YYYY-MM-DD) - `countryName` (may be ID or label, we normalize in frontend) - `cityName` - `locationName` - `timeZone`

Get Countries List

`GET /api/msme/couuntrylist` - Auth required - Returns array of `{ countryId, country_name }` - Used to hydrate Country dropdown.

Get States List

`GET /api/msme/statelist?countryId={id}` - Auth required - Returns array of `{ stateId, name }` - Called dynamically when Country changes. - Populates the State dropdown.

Update Personal Data

`POST /api/setting/personaldata` - Auth required - Body: - `firstName` (2–50 chars, required) - `lastName` (2–50 chars, required) - `gender` ("Male" | "Female", required) - `dob` (YYYY-MM-DD, required) - `countryName` (country ID string, required) - `cityName` (string, required) - `locationName` (string, required) - `timeZone` (optional) - Response: `{ success, message }`

3.3 Validation rules & UX states

- All required fields marked with `*`.
- DOB dropdowns → converted to `YYYY-MM-DD` before submit.
- Country must be chosen first; only then States are loaded. If states API call fails, we still allow manual city text but highlight "State list unavailable".
- Gender is required, binary right now. (If/when we expand gender options, Personal Data Popup spec must be revisited + policy updated.)
- We SHOULD validate minimum age (ex: 18+) if there is an age restriction on certain experiences.
- If under min age, frontend should block submit, show helper text.
- This logic aligns with safety/eligibility rules and helps with legal compliance.

Failure handling: - `/api/msme/couuntrylist` 401 → force re-auth before editing. - `/api/msme/statelist` 400/404 → show empty dropdown with helper "No states available". - `/api/setting/personaldata` 400/422 → - Keep modal open - Surface per-field errors inline - Do NOT lose partially edited form state - `/api/setting/personaldata` 500 → toast "We couldn't save your changes. Please try again." and console log.

3.4 Post-save behavior

- On success:
- Toast: "Your personal data was updated"
- Close modal
- Refresh parent view (Settings > Personal Data card now shows new values)
- Optionally refresh profile header / sidebar (name, city) without reload

3.5 RAG / Connie behavior for Personal Data

Connie should: - Be able to explain what each field is used for ("Your work city is used to match you to nearby assignments") - Warn that DOB is used for eligibility checks and certain legal protections - Clarify that users can update mistakes themselves through "Settings > Personal Data > Edit" - Walk the user through what to click, BUT: - Connie must NOT collect DOB, gender, or location in chat and claim it's saved. It can draft instructions but cannot pretend to execute the update. - If user asks "Why do you need my DOB?": Connie should answer from RAG using policy language around safety / eligibility / legal compliance.

---

4. Cross-cutting Compliance / Audit / Security

4.1 Consent trail

- Every `POST /api/user/privacy-consent` write produces:
- `consentId`
- `policyVersion`

- `acceptedAt`
- `ipAddress`
- `userAgent`
- We MUST retain this for audit + regulator inquiries.
- `GET /api/user/consent-history` exposes that trail back to the end user for transparency.

4.2 Grievance workflow

- `/api/support/grievance` creates a ticket like `GRV-2025-001`, with SLA metadata (ex: `48 hours`).
- This satisfies regulatory requirement (DPDPA etc.) for an accessible grievance channel.
- Connie RAG must surface:
- How to raise a privacy grievance
- Expected response time
- That they can do this even without logging in

4.3 Account / session safety

- Settings page also exposes recent login sessions (IP, browser, timestamp) via `/settings/` main account tab.
- This is indirectly privacy-related: user can spot suspicious sessions.
- We should keep that list visible in "My Account" tab (separate spec), but Connie should know it exists so it can say "You can review recent login sessions in Settings > My Account" if user asks "Has my account been accessed somewhere else?".

4.4 Data minimization / visibility

- Privacy Settings lets the user:
- Limit profile discoverability in search engines
- Hide activity/contacts from strangers
- Disable public mentions
- Mark themselves NOT open to recruiter outreach
- These toggles enforce *least exposure by choice*, which backs up statements in Privacy Policy about "user control".

4.5 Retention / access / deletion (policy-facing)

- Privacy Policy content (from `/api/legal/privacy-policy`) MUST include:
- How long we retain profile data and activity logs
- How a user may request correction or deletion
- How to escalate if they believe data is misused
- Connie will answer any "delete my data" / "how do I close my account" queries by:
- Guiding user to Settings → My Account → Deactivate / Permanently delete account
- OR telling them they can open a privacy grievance ticket asking for erasure
- Connie must **not** claim it has deleted anything itself

---

5. Frontend states & edge UX we MUST ship

5.1 Offline / API fail states

- Privacy Policy:
- If `/api/legal/privacy-policy` fails → show bundled static policy + small banner "Some content may be out of date".
- Privacy Settings:
- If `/api/settings/privacy` fails → render skeleton rows in read-only mode with banner error.
- Personal Data Popup:
- If countries/states fail loading → allow manual entry for city/location and warn "Some location lists are temporarily unavailable".

5.2 Authentication fallbacks

- Any `401` on protected endpoints immediately triggers re-auth modal / redirect.
- Never silently swallow 401 because that creates illusion of "saved" when nothing was saved.

5.3 Rate limiting / abuse controls

- `/api/support/grievance` can 429 → we must surface friendly cooldown text so legit users don't think we're ignoring them.
- Connie should also echo throttling rules instead of encouraging spam.

5.4 Accessibility / clarity

- All required fields: visually marked with `*` and explained in helper text.
- For DOB and gender (sensitive fields): include short trust copy like "We use this to protect your account and match eligibility."
- For privacy toggles: use human wording, e.g. "Who can see your Activity tab?" instead of purely technical labels.

---

6. Connie (AI Assistant) RAG Requirements for Privacy / Data Control

6.1 What goes into RAG index

We must ingest and keep up to date: 1. Latest Privacy Policy HTML + metadata from `/api/legal/privacy-policy` (including version, effectiveDate, compliance list) 2. FAQ-style explanations of: - Contacts tab visibility / Activity visibility / Search engine visibility / Mentions / Recruiter outreach - What happens if you reset to defaults 3. Personal Data Popup field purposes, validation requirements, and update flow 4. Grievance process + SLA + ticket format (`GRV-2025-001` style) 5. Account deletion / deactivation language from Settings → My Account 6. Session audit info (IP, browser, timestamp) to answer "Was someone else logged in?"

6.2 How Connie must answer

- Always ground answers in the ingested text. No hallucinated legal claims.

- Always tell user *where in product UI* they can act. Example:
- "You can update your gender and city by going to Settings > Personal Data and clicking Edit."
- Connie must never claim it already changed or saved anything.
- Connie can draft what the user will send in grievances ("Here's a template you can send about data deletion"), but should clarify: "You'll still need to submit this via the privacy grievance form."
- Connie should surface version info when talking about the Privacy Policy:
- "Current Privacy Policy is version 2.1, effective 15 Jan 2025."
- This gives legal a provable audit trail of what the user was told.

6.3 Safety boundaries

- If user asks something outside of policy scope (e.g. "Can you sell my personal data to X?"), Connie must:
- Check RAG for any sharing/third-party language.
- If not explicitly allowed, respond conservatively: "According to our Privacy Policy, ConnecWrk describes how your data may be processed and shared. I don't see language that says we sell your personal data for unrelated marketing. For more details you can read the Privacy Policy (version X.X) or raise a privacy grievance."
- If user appears to be a minor / under 18:
- Connie should advise limiting visibility (Activity tab `none`, etc.) and encourage involving a guardian if there's a safety concern, without giving legal advice.

---

7. Open items / next steps

- Add `age gating rule` (min age / parental consent) to Privacy Policy content so Connie can answer age-related compliance questions cleanly.
- Add multilingual support:
- `/api/legal/privacy-policy?lang=hi-IN` (future) should also be ingested so Connie can answer in Hindi when user is in Hindi locale.
- Add right-to-access / right-to-delete standard text blocks so Connie can explain "How do I download my data?" and "How do I permanently delete my account?" in a compliant, consistent way.
- Confirm retention/disclosure language for:
- Recruiter messages
- Assignment bids / proposals
- Uploaded resumes / portfolios So Connie can answer "Who can see my resume if I apply to a job?".

---

8. Summary

- **Privacy Policy page**: legal source of truth (public, versioned, consent logged).
- **Privacy Settings page**: user's control center for visibility and discoverability.
- **Personal Data Popup**: user's edit surface for identity, demographics, and location.
- **APIs** back these with:
- `/api/legal/privacy-policy` (content)
- `/api/user/privacy-consent` + `/api/user/consent-history` (audit trail)
- `/api/settings/privacy` (visibility preferences)
- `/api/setting/personaldata` (profile corrections)

- `/api/support/grievance` (escalations)
- **RAG/Connie** must stay in sync with all of the above to:
- answer accurately,
- guide users to in-product controls instead of faking changes,
- surface legal version + effective date,
- and always offer escalation via grievance ticket if the question is about rights, misuse, or data removal.