Datifyy - PRD

Absolutely! Here's a well-structured **Product Requirements Document (PRD)** introduction section for the **Datifyy Web App**, written in the style of a senior Product Manager for a dating-tech startup with a high emphasis on UX, scalability, and consistency.

# **📝 Product Requirements Document (PRD)**

## **📍 Project: Datifyy – Verified Dating Web Application**

## **🗓️ Phase: Initial MVP + Admin Tooling + Responsive Frontend**

### **🧭 Overview**

**Datifyy** is a modern, responsive dating web platform built to provide **verified, curated, and meaningful** dating experiences for users. Unlike typical swipe-based dating apps, Datifyy emphasizes **human-curated matchmaking**, **partner preference alignment**, and **real-time engagement** through guided video and in-person date setups.

We focus on creating **trust-based interactions** where every user is verified, and every match is intentional.

### **🎯 Vision & Core Objectives**

* Enable users to discover and connect with curated matches based on deeply aligned partner preferences.
* Build a robust admin interface to allow for match curation, feedback analysis, and communication workflows.
* Maintain an engaging and consistent UX across **mobile**, **tablet**, and **desktop** devices.
* Uphold **trust, safety, and verification** as core pillars of the platform.
* Ensure high-quality user data collection to power algorithmic and admin-led matchmaking.

### **📱 Platform UX Guidelines (Critical)**

#### **✅ Responsive by Default**

* All views must be **fully responsive** and tested across:  
  + Mobile (iOS + Android breakpoints)
  + Tablets
  + Desktops (HD to widescreen)
* Every component should **scale fluidly** and **retain visual clarity** and functionality.

#### **✅ Unified Design System**

* Chakra UI as the component library with **custom theming for Datifyy**.
* Consistent typography, spacing, form fields, modals, cards, and interaction feedback.
* All admin and user views follow the **same design language**, ensuring familiarity and reduced cognitive load.

#### **✅ Intuitive & Addictive Experience**

* Flow of filling user information (including profile & partner preferences) should feel **effortless and rewarding**.
* Progressive disclosure, autosave, and guidance patterns encouraged.
* Use micro-interactions and nudges to keep users engaged throughout setup.

### **🧑‍💻 User Journey – Key Screens & States**

#### **🔐 Authentication Flow**

* When a user visits **datifyy.com**, the header shows:  
  + **Login / Signup buttons** (if not authenticated)
  + **Current City/State** detection (with dropdown to change)
  + **Language Selector** (multi-lingual support planned)
* Responsive navbar behavior:  
  + Collapsible on mobile, persistent on desktop

#### **🏠 Landing Page**

* Communicates **who we are**, **why Datifyy is different**, and **how the platform works**.
* Encourage new users to **start building their profile** or learn more about our **verified speed dating** and **curated date formats**.

### **🧍User Profile Setup**

* **Step-by-step guided experience** to collect key details:  
  + Personal information (name, gender, DOB, etc.)
  + Identity verification (govt ID, workplace, email)
  + Preferences and lifestyle (diet, religion, education, habits)
  + Photos & bio
* Emphasis on building **trust, safety**, and match quality through depth of profile.

### **💞 Partner Preferences Page**

* A **dedicated experience** where users submit detailed preferences for their ideal match.
* For each field, users specify:  
  + Their preference (e.g., education level, salary range)
  + Match strength:  
    - **Must Have**
    - **Nice to Have**
    - **Doesn’t Matter**
* This structured input powers **matching algorithms and admin curation**.
* Smooth UX with:  
  + Autocomplete fields
  + Step indicators
  + Conditional sections based on gender or orientation

### **🔁 Mutation UX & Error Handling**

* All mutation actions (e.g., form submit, delete, update) must:  
  + Trigger a **confirmation modal**
  + Handle **backend JSON error messages** gracefully
  + Show **user-friendly toasts or alerts** (e.g., “Something went wrong. Please try again.”)
* Optimistic UI updates where appropriate.

### **🧑‍💼 Admin Interface (Brief Preview)**

* Admins are **central to the Datifyy experience**.
* They curate matches, manage sessions, trigger communication workflows, and ensure platform integrity.
* Admin interface adheres to the same design system and UX quality.
* Built with enterprise-grade controls (e.g., filters, bulk actions, data insights).

Let me know when you're ready to move into the **next section** of this PRD — for example:

* Profile field schema
* Partner preference schema
* API design
* UX flows or wireframes
* Admin panel specs
* Feedback loop integrations  
   I'll continue building it with the same clarity and production-readiness.

Here’s a well-organized **PRD section** for the **Availability Management** feature, written in a professional product document style as part of the Datifyy platform:

## **📆 Availability Management – User Scheduling Module**

### **🧩 Part of: Datifyy – Verified Dating Experience Platform**

### **🎯 Objective**

To enable users to submit, view, and manage their **availability for dates over the upcoming 7 days**, allowing admins to match and schedule curated dates efficiently. This ensures real-time matchmaking readiness, improves success rates, and facilitates structured, mutual interactions.

## **🔑 Core User Stories**

### **1. ✅ Submit Availability**

* Users can select **available date & time slots** for the next 7 days.
* Each slot includes:  
  + 📅 Day (e.g., Friday, Saturday, etc.)
  + 🕒 Time Block:  
    - Morning (8AM–12PM)
    - Afternoon (12PM–4PM)
    - Evening (4PM–8PM)
    - Night (8PM–11PM)
  + 🧍 Mode: **Online** or **Offline**
    - If **Offline**:  
      * Let users choose a **venue type** (e.g., Café, Park, Co-working lounge)
      * Future enhancement: allow them to select from curated partner venues
* UX:  
  + Show slot selection via **interactive calendar view** or time-grid
  + Disable slots already booked
  + Allow multi-slot selection in one go
  + Optimistic UI: visual confirmation before submission

### **2. ✏️ Edit or Cancel Availability**

* Users can view their **submitted availability** at any time  
  + Display grouped by day, then time block
  + Tag for online/offline
* Users can:  
  + **Edit** slot mode (online/offline)
  + **Delete** slot
* When **cancelling an availability slot**:  
  + If a **date is already scheduled** in that slot:  
    - ❌ The date is auto-cancelled
    - 📧 Trigger email + in-app notification for **both users**
    - Log admin-side audit of cancellation cause

### **3. 🔔 Reminders & Nudges**

* If a user has:  
  + 0 or ≤1 upcoming availability for **next weekend**, trigger:  
    - 📩 Email reminder every **Wednesday morning** (local time)
    - 🔔 In-app notification when they next open the platform
    - Message copy: *“Your weekend’s looking empty! Add availability to get matched.”*
* Part of an **engagement automation engine** to boost active participation.

### **4. 🚫 Slot Conflict Detection**

* While submitting availability:  
  + **Already booked slots (confirmed dates)** are **disabled** and clearly marked.
  + Users cannot override those.
* Avoid overbooking or double submissions.

### **5. 💻 Responsive Design & UX**

* Mobile-first with tablet + desktop support
* Calendar grid must feel **natural and tactile** on mobile (tap to select)
* Form submission always confirms with a **modal**
* Backend returns JSON errors – frontend gracefully parses and displays meaningful errors

### **🛠️ Admin Impact**

* Availability data is consumed by the **Admin Panel** to:  
  + Show upcoming available slots for match curation
  + Prevent conflicts with pre-booked dates
  + Understand user participation patterns

### **🔐 Privacy & Compliance**

* All availability data is tied to a verified user
* Slot times are never exposed to other users
* Cancellations are only processed via confirmed user actions

Absolutely. Below is the **Charges & Business Model PRD section** for Datifyy, designed for both global and India-first monetization strategies. It balances **user experience**, **revenue generation**, and **trust-building**, while offering scalable monetization hooks.

## **💰 Charges & Business Model – Monetization Strategy for Datifyy**

### **🎯 Goal**

To build a **sustainable and user-aligned monetization model** that respects user trust and delivers clear value, while maximizing revenue through **Love Tokens**, **freemium tiering**, and **premium date curation services**. All flows must be intuitive, mobile-optimized, and respectful of relationship-building intentions.

### **🧑‍🤝‍🧑 Dating Credits – "Love Tokens"**

#### **🪙 What Are Love Tokens?**

A virtual currency used within Datifyy to:

* **Accept and confirm curated dates** (especially offline dates)
* **Get priority in match curation**
* **Redeem for special experiences** (e.g., themed dates, dating events)

#### **🎁 Free Credit Allotment**

* Upon signup, users receive:  
  + X free tokens for **online dates**
  + Y free tokens for **offline dates**
* Encourages early engagement and incentivizes completion of profile and partner preferences

### **🛒 Purchasing Love Tokens**

#### **Access Points:**

* **Profile Menu** → “Buy Love Tokens”
* **Header Icon** → with token balance + CTA ("Top up")
* **Date Curation Page** → if user doesn't have enough tokens, prompt contextual CTA

#### **UX Design:**

* Modal with token packs:  
  + 10 tokens – ₹49
  + 50 tokens – ₹199
  + 100 tokens – ₹349
* Offer promo discounts (e.g., +10% bonus tokens)
* Payment methods (for Indian users):  
  + UPI (PhonePe, GPay, Paytm)
  + Credit/Debit Cards
  + Netbanking
  + Wallets
  + (Global: Stripe, Apple Pay, Google Pay)

### **📦 Billing Page**

Users can visit the **Billing** page to:

* View token balance
* See **purchase history** (token packs, amount, date/time)
* Download invoices (PDF)
* Track **token usage logs** (e.g., used for an offline date on July 21)

### **🔁 Usage & Revenue Flow**

| **Feature** | **Tokens Required** | **Monetization Purpose** |
| --- | --- | --- |
| Accept online curated date | 1 token | Engage |
| Accept offline curated date | 3 tokens | Premium intent |
| Join verified speed dating event | 5–10 tokens | Paid event |
| Re-match with a previous partner | 2 tokens | Emotional connection |
| Boost profile visibility | 5 tokens | Revenue & visibility |

## **🧠 India-First Monetization Strategy**

### **🇮🇳 What Works in Indian Market:**

#### **1. Freemium with Earnable Rewards**

* Allow users to earn tokens by:  
  + Referring friends
  + Completing profiles
  + Giving feedback after dates
  + Participating in offline events

#### **2. Event-Based Revenue**

* Host curated local meetups (e.g., “Coffee & Connect” in major Indian cities)
* Charge via tokens or INR
* Partner with cafes and lounges to co-host and earn affiliate revenue

#### **3. Gifting & In-App Purchases**

* Allow users to send **gifts** (virtual flowers, hearts) using tokens
* Use **regional pricing** and festive promotions (Diwali, Valentine's Day, etc.)

#### **4. Premium Membership Tier (Post-MVP)**

* Monthly subscription for:  
  + Unlimited curated dates
  + Profile insights & visibility boost
  + Chat concierge with admin advice
  + Priority support
  + Advanced filtering (salary, location, verification level)

## **🔐 Trust, Transparency, & UX Guidelines**

* Always show **token balance** in profile
* Clear breakdown of:  
  + What costs tokens
  + How they’re spent
* Every **purchase** confirmed via **modal dialog** with cancellation option
* Graceful error handling for payment failures (JSON-based error display)

## **📊 Admin Insights (For Internal Use)**

Admins can:

* Track purchases, usage, and high-spending users
* Create custom offers (e.g., “10 tokens free for Valentine’s week”)
* Run cohort analysis on token-based engagement vs. free users

Would you like:

* Wireframe of token purchase modal or billing page?
* Component breakdown using Chakra UI?
* API contract for token management and billing history?

Ready to build the tech next.

## **💘 Date Curation Page – User-Facing Experience**

**For Datifyy – The Verified Dating Platform**

### **🎯 Objective**

The *Date Curation* page serves as the personalized match and date management hub for users. It is central to building trust and engagement by helping users track curated dates, view details, take actions, and interact with Datifyy's matchmaking agents in a conversational, humanized flow.

## **🧩 MVP Features**

### **1. Entry Points (High Visibility)**

* Top navigation bar item: "Your Dates"
* Home/Landing Page prompt (if user is logged in) — highlight if there are active matches curated by admins.  
  + Ex: “You’ve got a curated date! 🎯 Tap to view.”

### **2. Upcoming Dates Section**

* If an admin has curated any upcoming date(s), the user will see:  
  + **Date Details Card**:  
    - Match’s Name (First name only or with initial)
    - Date & Time
    - Mode: Online (with secure video link) or Offline (Google Maps Location)
    - Dress code recommendation / event type (optional)
    - Custom note from admin (if any)
  + Action Buttons:  
    - ✅ **Accept** (if needed)
    - ❌ **Cancel** → must select a reason (No time, Not interested, Emergency, etc.)  
      * Warning: *“Too many cancellations will impact your profile score and future matches.”*
  + **Send Reminder Toggle** (default ON): Email + WhatsApp 24 hours before the date.

### **3. Date History & Progression**

* Dedicated section: “Past Dates”  
  + List of previously completed dates with:  
    - Name of match
    - Date & Time
    - Status: Attended / Cancelled
    - Feedback summary (if submitted)
* If going on a **2nd or 3rd date with the same person**, show:  
  + Previous interaction dates
  + Feedback exchanged
  + Match rating evolution over time
  + Admin's brief comment (optional): “This match looks promising 💖”

### **4. Profile Impact Warning**

* If the user cancels frequently or last-minute:  
  + Show UI alert: *“Frequent cancellations negatively impact your trust score.”*
  + Reduce profile visibility algorithmically (optional)
  + Educate the user: Why reliability matters in real-world dating.

### **5. Match Compatibility Score**

* Show **Matching Score** with each date (as a percentage or star rating):  
  + Based on shared preferences, lifestyle alignment, values, etc.
  + Add conversion likelihood: *“79% chance this could lead to a meaningful relationship.”*

## **🔮 Post-MVP: Conversational & Messaging Enhancements**

### **6. Matchmaker Chat (Agent Interface)**

* Embed a secure, in-app chat with a Datifyy Agent (e.g., “Your Dating Concierge”):  
  + Natural, conversational UX (no personal data sharing)
  + Use cases:  
    - Ask questions about date
    - Share concerns or preferences
    - Get tips (dress, venue, talking points)
  + Agent can suggest additional matches or confirm feedback.

### **7. Live Match Chat (with Date Partner)**

* Once both confirm and are within 24 hours of the date, enable:  
  + Secure, private in-app chat (without sharing contact details)
  + Features:  
    - Text, emoji, and formatted message support
    - Photo sharing (with moderation)
    - Typing indicators, "live" online presence badge
    - Chat history

## **🔐 Privacy & Safety Notes**

* All interactions are **moderated and encrypted**.
* No personal contact details are ever shared.
* Admins and AI moderation flag inappropriate behavior automatically.

## **📱 Notifications & Automation**

* **Email + WhatsApp reminders** sent 24 hours before each confirmed date
* Post-date feedback request triggered automatically
* Cancellation alerts with soft reminders about community behavior expectations

Absolutely. Below is the **ADMIN PANEL – Full PRD** for **Datifyy**, designed from the perspective of an experienced Product Manager building scalable, insightful, and secure internal tools for matchmaking operations, analytics, and monetization tracking.

# **🛠️ Datifyy Admin Dashboard – Product Requirements Document**

## **🔐 Admin Access**

* **Admin Login Only**
  + No signup or forgot password flow
  + Auth restricted to pre-approved admin credentials
* Secure route: /admin
* Admin sessions should expire after 24 hours or inactivity

## **🏠 Admin Home – Dashboard Overview**

A high-visibility snapshot of platform growth, activity, and operations across timeframes. All metrics include comparisons and trendlines (vs last week, month, quarter).

### **📊 Key Metrics (With Graphs & Comparisons)**

| **Metric** | **Timeframes** |
| --- | --- |
| 📈 Total Users | Lifetime + growth trend vs last week/month/3 months |
| 🟢 Today’s Active Users | Real-time + trend line |
| 💑 Total Dates Setup | All-time + daily + trend |
| ❌ Total Dates Cancelled | With cancellation reason trends |
| 📅 Total Open Slots for Upcoming Weekend | Segmented by gender, city |
| 🙋‍♀️ Total Users with Upcoming Slots | Week-on-week change |
| 👩‍👦 Total Males / Females | By city, state, country |
| 🌐 Dates Scheduled (Next Week) | With breakdown by online/offline & location |
| 📍 Dates by City & Country | Top performing cities by dating activity |

📉 Include line/bar charts for growth  
 🗺️ Heatmaps for location activity  
 📅 Trends in engagement around weekends or holidays

## **❤️‍🔥 Curate Dates – Matchmaking Workflow**

### **User Table View (List of Participants)**

* Tabular layout: sortable, filterable, paginated
* **Columns**:  
  + Name, Email
  + Active/Inactive/Deactivated
  + Last Active
  + Total Dates Attended
  + Profile Score
  + Verified (ID / Work)
  + Submitted Availability (Yes/No)
  + City, Country
  + Gender
  + Job & Salary

#### **🔍 Filters & Sorting**

* Filters:  
  + City / Country
  + Gender
  + Verification status
  + Job / Salary bracket
  + Submitted availability (upcoming 7 days)
* Sorting:  
  + Default: Profile Score (DESC)
  + Others: Last Active, Total Dates, etc.

### **🧮 Row Expansion for Match Suggestions**

* On selecting a user row:  
  + Show a list of **potential opposite-gender matches**
  + Each suggestion shows same key profile fields
  + **Status tags** if:  
    - Already has a date this weekend
    - Has past date with this user (click to expand past feedback)
* Admin can:  
  + Select one or more matches
  + Click **“Curate Date”**
  + Confirm via modal: set time, mode (online/offline), location
  + Automatically trigger email/WhatsApp invites

📬 Admin can also send email templates like:

* Forgot password
* Event reminders
* Feedback requests

## **📅 Curated Dates Management**

### **Calendar or Tabular View**

* See all upcoming curated dates
* Filter by:  
  + Date range
  + City / Country
  + Online vs Offline

### **Actions:**

* Cancel Date (modal with reason, optional message)
* Trigger feedback email (manually or scheduled)
* View feedback from both users
* Trigger reschedule (admin suggested time)
* Send follow-up email or token refund

## **💸 Revenue & Purchase Tracking (World-Class View)**

### **Summary Section**

* Total Revenue (all-time)
* Revenue by:  
  + Week (vs last week)
  + Month (vs last month)
  + 3 Months
* Paid Users count
* Revenue by Date Type:  
  + Online vs Offline
* Token Packs sold (volume + ₹ amount)

### **Graphs & Breakdowns**

* Line/Bar chart: Revenue over time
* Pie chart: Paid vs Free Dates
* Heatmap or table: Revenue by **City, State, Country**
* Table: Top 10 revenue cities
* Revenue split by:  
  + Platform (Web, Mobile)
  + Event Type (Speed Dating, Curated 1:1, etc.)

## **🔔 Real-Time Notifications (Slack + Email)**

Set up automation to notify Admin team for important events:

| **Event** | **Slack** | **Email** |
| --- | --- | --- |
| ✅ User purchases token pack | ✅ | ✅ |
| 💬 User gives feedback | ✅ | ✅ |
| ❌ User cancels a date | ✅ | ✅ |
| 🔁 User requests reschedule | ✅ | ✅ |
| 💰 Revenue crosses key thresholds | ✅ | ✅ |

All notifications to be configured via Slack webhook + mailer backend (e.g., Mailgun, Sendgrid)

## **🔐 Security & Role Management**

* Role: admin
* Add ability later for super admin, moderator
* Admin audit log:  
  + Who curated which date
  + Who cancelled/rescheduled
  + Token refund history

## **🔍 Future Add-ons**

* Admin search: Find user or date by ID/email
* Export CSV of metrics
* A/B testing: Email template performance
* Admin comments or notes per user

Would you like:

* Wireframes for this admin panel?
* DB schema for admin\_curated\_dates, revenue\_logs, feedbacks, etc.?
* Suggested Chakra UI component breakdown for the dashboard graphs and tables?

Let me know, and I’ll build it all accordingly.

/\*\*

\* ---------------------------------------------

\* Types for Modern Dating App User Profile

\* Curated from Bumble, Hinge, Tinder, OkCupid, etc.

\* ---------------------------------------------

\*/

/\*\* Gender identity \*/

export enum Gender {

Male = 'male',

Female = 'female',

NonBinary = 'non-binary',

Transgender = 'transgender',

Other = 'other',

}

/\*\* Drinking habits \*/

export enum DrinkingHabit {

Yes = 'yes',

No = 'no',

Occasionally = 'occasionally',

}

/\*\* Smoking habits \*/

export enum SmokingHabit {

Yes = 'yes',

No = 'no',

Occasionally = 'occasionally',

}

/\*\* Relationship status \*/

export enum RelationshipStatus {

Single = 'single',

Divorced = 'divorced',

Widowed = 'widowed',

Separated = 'separated',

}

/\*\* Children preference \*/

export enum ChildrenPreference {

Yes = 'yes',

No = 'no',

Undecided = 'undecided',

}

/\*\* Exercise habit \*/

export enum ExerciseHabit {

Regularly = 'regularly',

Sometimes = 'sometimes',

Never = 'never',

}

/\*\* Diet type \*/

export enum Diet {

Vegetarian = 'vegetarian',

Vegan = 'vegan',

NonVegetarian = 'non-vegetarian',

Eggetarian = 'eggetarian',

Halal = 'halal',

Kosher = 'kosher',

Other = 'other',

}

/\*\* Verification types \*/

export enum VerificationType {

GovtId = 'govt\_id',

Email = 'email',

WorkEmail = 'work\_email',

Manual = 'manual',

}

/\*\* Connection mode preference \*/

export enum ConnectionMode {

Chat = 'chat',

Call = 'call',

Video = 'video',

InPerson = 'in-person',

}

/\*\* Intent / Match Goals \*/

export enum MatchIntent {

LongTerm = 'long-term',

ShortTerm = 'short-term',

Casual = 'casual',

Friendship = 'friendship',

Marriage = 'marriage',

}

/\*\*

\* Normalized location schema for proximity matching and geographic filters.

\* Supports city, state, country, lat/lng, and geohash for spatial queries.

\*/

export interface UserLocation {

/\*\* Display-friendly location name (e.g., "San Francisco, CA, USA") \*/

displayName: string;

/\*\* ISO 3166-1 alpha-2 country code (e.g., "US") \*/

countryCode: string;

/\*\* Country full name (e.g., "United States") \*/

country: string;

/\*\* State or region (e.g., "California") \*/

state?: string;

/\*\* City or locality (e.g., "San Francisco") \*/

city?: string;

/\*\* Neighborhood or sublocality (optional) \*/

subLocality?: string;

/\*\* Latitude for proximity matching \*/

lat: number;

/\*\* Longitude for proximity matching \*/

lng: number;

/\*\* Optional: geohash for spatial indexing \*/

geoHash?: string;

}

/\*\*

\* Represents one educational qualification or milestone.

\*/

export interface EducationEntry {

/\*\* Type of education \*/

type: 'undergraduate' | 'postgraduate' | 'phd' | 'diploma' | 'other';

/\*\* Institution/college name \*/

institution: string;

/\*\* Major or field of study \*/

major?: string;

/\*\* Graduation year (4-digit format) \*/

graduationYear?: number;

/\*\* Country or location of the institution \*/

location?: string;

}

/\*\*

\* Final user profile schema for dating apps.

\*/

export interface UserProfile {

/\*\* Unique user ID \*/

userId: string;

/\*\* First name \*/

firstName: string;

/\*\* Last name (optional, private) \*/

lastName?: string;

/\*\* Email (used for auth + communication) \*/

email: string;

/\*\* User's gender \*/

gender: Gender;

/\*\* Interested in genders \*/

interestedIn: Gender[];

/\*\* Date of birth (YYYY-MM-DD) \*/

dateOfBirth: string;

/\*\* Age preference for matching \*/

agePreference: {

min: number;

max: number;

};

/\*\* Height in centimeters \*/

heightCm?: number;

/\*\* Occupation / job title \*/

occupation?: string;

/\*\* Company / workplace \*/

company?: string;

/\*\* Location (with geospatial data) \*/

location: UserLocation;

/\*\* Hometown (optional) \*/

hometown?: string;

/\*\* Languages spoken \*/

languages?: string[];

/\*\* Religion or beliefs \*/

religion?: string;

/\*\* Ethnicity \*/

ethnicity?: string;

/\*\* Drinking habits \*/

drinking?: DrinkingHabit;

/\*\* Smoking habits \*/

smoking?: SmokingHabit;

/\*\* Children preference \*/

childrenPreference?: ChildrenPreference;

/\*\* Marital status \*/

relationshipStatus?: RelationshipStatus;

/\*\* Zodiac sign \*/

zodiacSign?: string;

/\*\* Political views \*/

politics?: string;

/\*\* Exercise habits \*/

exercise?: ExerciseHabit;

/\*\* Diet type \*/

diet?: Diet;

/\*\* Pet preferences or pets owned \*/

pets?: string;

/\*\* Short bio (self-summary) \*/

bio: string;

/\*\* Fun prompts (e.g., "I'm known for...") \*/

prompts?: Array<{

question: string;

answer: string;

}>;

/\*\* Profile photos (URLs or media keys) \*/

photos: string[];

/\*\* Instagram handle \*/

instagramHandle?: string;

/\*\* Spotify top artists (if connected) \*/

spotifyTopArtists?: string[];

/\*\* LinkedIn profile (optional for verification) \*/

linkedInUrl?: string;

/\*\* Government ID or work email verified \*/

isVerified: boolean;

/\*\* Type of verification \*/

verificationType?: VerificationType;

/\*\* Last seen or active timestamp \*/

lastActiveAt: string;

/\*\* What the user is looking for \*/

lookingFor: MatchIntent[];

/\*\* Distance preference in kilometers \*/

distancePreferenceKm: number;

/\*\* Is profile manually reviewed/approved (e.g., Aisle style) \*/

isProfileApproved: boolean;

/\*\* Display badges (e.g., “Vaxxed”, “Dog lover”) \*/

badges?: string[];

/\*\* Preferred mode of communication \*/

preferredConnectionMode?: ConnectionMode;

/\*\* Consent to be shown/matched \*/

consentToMatchmaking: boolean;

/\*\* Education history (can include undergrad, postgrad, PhD, etc.) \*/

educationHistory?: EducationEntry[];

/\*\* Most recent or simple education summary (optional) \*/

education?: string;

}

PArtner preference  
/\*\*

\* ----------------------------------------------------

\* Partner Preferences Schema for a Dating Application

\* Inspired by Bumble, Hinge, Tinder, Aisle, OkCupid

\* Designed to support match priorities (must-have, nice-to-have)

\* ----------------------------------------------------

\*/

/\*\* Importance level for matching \*/

export enum MatchPriority {

MustHave = 'must\_have',

NiceToHave = 'nice\_to\_have',

DoesntMatter = 'doesnt\_matter',

}

/\*\* Preferred gender(s) to match with \*/

export enum PreferredGender {

Male = 'male',

Female = 'female',

NonBinary = 'non-binary',

Transgender = 'transgender',

Other = 'other',

All = 'all',

}

/\*\* General lifestyle yes/no/occasionally values \*/

export enum LifestyleValue {

Yes = 'yes',

No = 'no',

Occasionally = 'occasionally',

Doesn’tMatter = 'doesn’t\_matter',

}

/\*\* Intentions or relationship goals \*/

export enum RelationshipGoal {

LongTerm = 'long-term',

ShortTerm = 'short-term',

Casual = 'casual',

Friendship = 'friendship',

Marriage = 'marriage',

OpenToAll = 'open-to-all',

}

/\*\* Education levels \*/

export enum EducationLevel {

HighSchool = 'high-school',

Bachelors = 'bachelors',

Masters = 'masters',

Doctorate = 'doctorate',

Diploma = 'diploma',

Other = 'other',

}

/\*\* Willingness to relocate for partner \*/

export enum RelocationPreference {

Yes = 'yes',

No = 'no',

Maybe = 'maybe',

}

/\*\* Body type preference \*/

export enum BodyTypePreference {

Slim = 'slim',

Athletic = 'athletic',

Average = 'average',

Curvy = 'curvy',

Doesn’tMatter = 'doesn’t\_matter',

}

/\*\* Astrological compatibility interest \*/

export enum AstrologyBelief {

Strong = 'strong',

Moderate = 'moderate',

None = 'none',

Doesn’tMatter = 'doesn’t\_matter',

}

/\*\* Political compatibility preference \*/

export enum PoliticalLeaning {

Left = 'left',

Center = 'center',

Right = 'right',

Apolitical = 'apolitical',

Doesn’tMatter = 'doesn’t\_matter',

}

/\*\* Work style compatibility \*/

export enum WorkStyle {

Remote = 'remote',

Hybrid = 'hybrid',

Onsite = 'onsite',

Doesn’tMatter = 'doesn’t\_matter',

}

/\*\*

\* A value with its assigned matching priority.

\*/

export interface PreferenceWithPriority<T> {

value: T;

priority: MatchPriority;

}

/\*\*

\* Full schema to store partner preferences with weighting

\* Enables AI-powered matchmaking based on core values and goals

\*/

export interface PartnerPreferences {

/\*\* Gender(s) user wants to match with \*/

preferredGenders: Array<PreferenceWithPriority<PreferredGender>>;

/\*\* Age range preference \*/

ageRange: {

min: number;

max: number;

priority: MatchPriority;

};

/\*\* Maximum distance in kilometers \*/

maxDistanceKm?: PreferenceWithPriority<number>;

/\*\* Preferred height range \*/

heightRangeCm?: {

min: number;

max: number;

priority: MatchPriority;

};

/\*\* Smoking habit preference \*/

smoking?: PreferenceWithPriority<LifestyleValue>;

/\*\* Drinking habit preference \*/

drinking?: PreferenceWithPriority<LifestyleValue>;

/\*\* Exercise frequency preference \*/

exercise?: PreferenceWithPriority<LifestyleValue>;

/\*\* Dietary preferences \*/

diet?: PreferenceWithPriority<

'vegetarian' | 'vegan' | 'non-vegetarian' | 'eggetarian' | 'halal' | 'kosher' | 'doesn’t\_matter'

>;

/\*\* Whether user wants children \*/

wantsChildren?: PreferenceWithPriority<LifestyleValue>;

/\*\* Whether user is okay if partner already has children \*/

hasChildren?: PreferenceWithPriority<LifestyleValue>;

/\*\* Religious preferences \*/

religion?: Array<PreferenceWithPriority<string>>;

/\*\* Ethnicity preferences \*/

ethnicity?: Array<PreferenceWithPriority<string>>;

/\*\* Languages spoken/preferred \*/

languages?: Array<PreferenceWithPriority<string>>;

/\*\* Caste preferences (region-specific) \*/

caste?: Array<PreferenceWithPriority<string>>;

/\*\* Preferred zodiac signs \*/

zodiacSigns?: Array<PreferenceWithPriority<string>>;

/\*\* Education level preferences \*/

educationLevels?: Array<PreferenceWithPriority<EducationLevel>>;

/\*\* Relationship goals (e.g., long-term, marriage) \*/

lookingFor: Array<PreferenceWithPriority<RelationshipGoal>>;

/\*\* Body type preference \*/

bodyType?: PreferenceWithPriority<BodyTypePreference>;

/\*\* Belief in astrology compatibility \*/

astrologyBelief?: PreferenceWithPriority<AstrologyBelief>;

/\*\* Political views compatibility \*/

politicalView?: PreferenceWithPriority<PoliticalLeaning>;

/\*\* Preferred work lifestyle \*/

workStyle?: PreferenceWithPriority<WorkStyle>;

/\*\* Income range in user’s local currency \*/

incomeRange?: {

min: number;

max: number;

priority: MatchPriority;

};

/\*\* Willingness to relocate for partner \*/

willingToRelocate?: PreferenceWithPriority<RelocationPreference>;

}