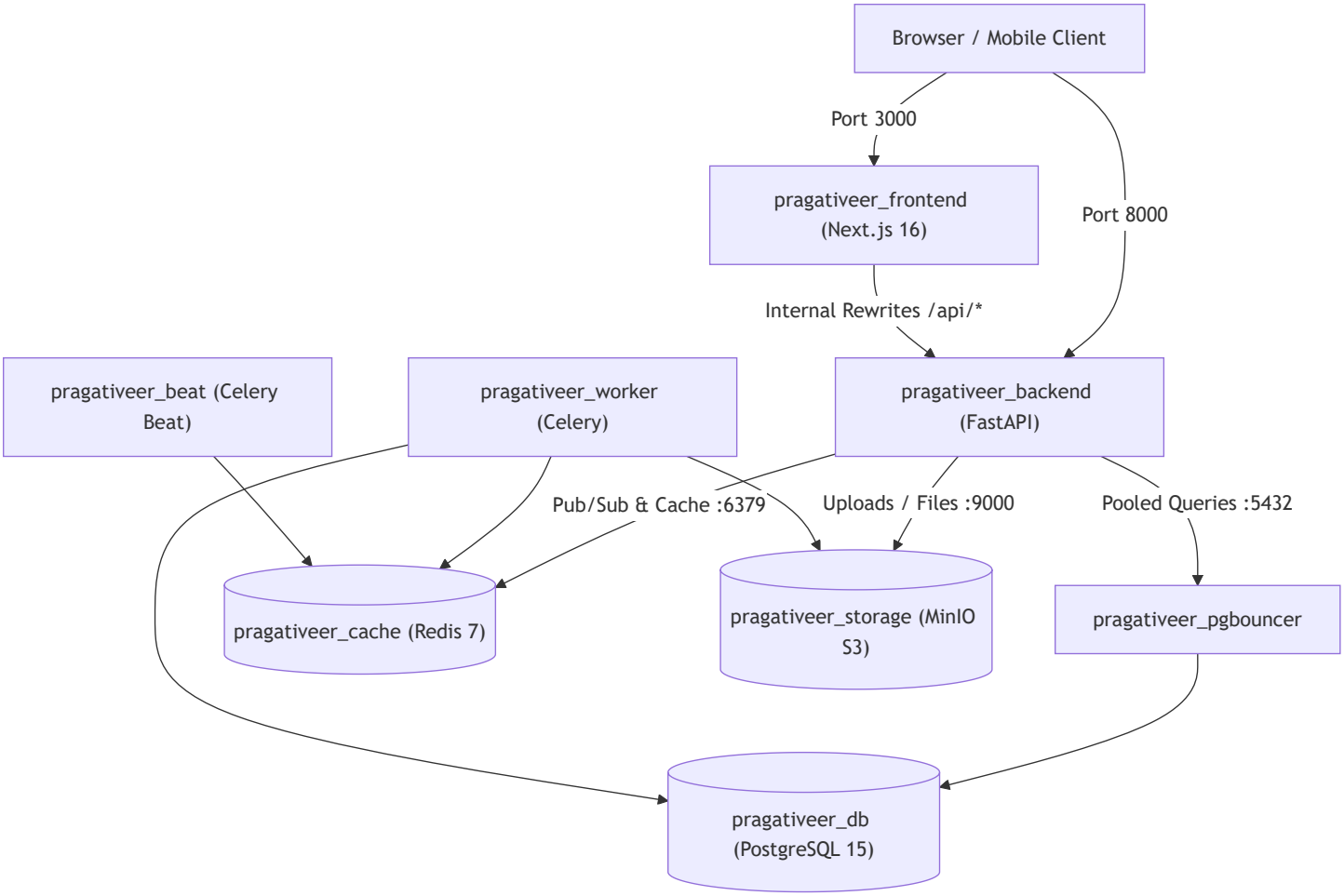
PragatiVeer v2 — Comprehensive API & Architecture Developer Documentation

> **Document Type:** Full Developer Reference & Technical API Specification
> **Target Audience:** Frontend Engineers, Backend Engineers, DevOps, QA Engineers
> **Last Verified:** September 2026
> **Backend Engine:** FastAPI (Python 3.11) + SQLModel (SQLAlchemy) + Celery + Redis
> **Frontend Engine:** Next.js 16 (React 19, TypeScript, TailwindCSS)
> **Database & Storage:** PostgreSQL 15 (PgBouncer connection pooler) + MinIO (S3-compatible)

1. System Architecture & Container Topology

PragatiVeer v2 operates as an 8-container orchestrated cluster via Docker Compose:



Port Mapping & Infrastructure Reference

Container Service	Image / Engine	Internal Port	Host Port	Role
`pragativeer_frontend`	Next.js 16 Standalone	3000	`3000`	UI, CBT Interface, Next.js Edge OG Image Generation
`pragativeer_backend`	Python 3.11 / Uvicorn	8000	`8000`	Core REST API, Auth, CBT Engine, AI Coach
`pragativeer_db`	PostgreSQL 15 Alpine	5432	`5432`	Relational storage (Users, Tests, Attempts, Mistakes)
`pragativeer_pgbouncer`	PgBouncer	5432	Internal	High-concurrency connection pooler for PostgreSQL
`pragativeer_cache`	Redis 7 Alpine	6379	`6379`	Response cache & Celery task message broker
`pragativeer_worker`	Celery Worker	—	Internal	Background PDF AI question parsing & exam analytics
`pragativeer_beat`	Celery Beat	—	Internal	Scheduled 5:00 AM IST study plan generation
`pragativeer_storage`	MinIO S3	9000 / 9001	`9000` / `9001`	PDF test uploads, question diagrams, user avatars

2. Authentication, Cookies & Request/Response Headers Overview

2.1 Next.js 16 Proxy Middleware Protection (`src/proxy.ts`)

Next.js 16 uses `src/proxy.ts` (modern convention replacing older middleware patterns) to govern edge route protection:

- **Cookie Name:** `auth_token`
- **Dashboard Routes (`/dashboard/:path*`):** Requires cookie `auth_token=<user_id_or_token>` . If absent, redirects to `/login` (HTTP 307 Temporary Redirect).
- **Admin Routes (`/admin/:path*`):** Requires cookie `auth_token=admin` . If mismatched or missing, redirects to `/login` (HTTP 307 Temporary Redirect).
- **Public Routes:** `/` , `/login` , `/signup` , `/about` , `/contact` , `/pricing` , `/cbt-exam/:path*` .

2.2 Global Header Standards Across APIs

All PragatiVeer v2 endpoints conform to strict HTTP/1.1 and HTTP/2 header conventions:

Common Request Headers

Header	Description	Required On
`Accept`	`application/json`, `text/event-stream`, or `image/png`	All API requests
`Content-Type`	`application/json` (or `multipart/form-data` for file uploads)	All `POST`, `PUT`, `PATCH` requests
`Cookie`	`auth_token=<uuid_or_admin>`	Protected routes (`/dashboard/*`, `/admin/*`)
`X-Razorpay-Signature`	HMAC-SHA256 hex digest computed with Razorpay Webhook Secret	`POST /api/billing/webhook/razorpay`
`Cache-Control`	`no-cache`	Streaming SSE endpoints

Common Response Headers

Header	Value / Usage	Endpoints
`Content-Type`	`application/json`	REST endpoints
`Content-Type`	`text/event-stream; charset=utf-8`	AI Tutor Streaming (`/api/ai-coach/chat/stream`)
`Content-Type`	`image/png`	Dynamic OG Image Generator (`/api/og/progress`)
`Cache-Control`	`no-cache, no-transform`	Real-time AI Coach streaming
`Connection`	`keep-alive`	Long-lived streaming connections
`X-Accel-Buffering`	`no`	Prevents Nginx/Reverse Proxy chunk buffering

2.3 Tier-Based Gating (`require_tier`)

FastAPI dependency `require_tier(allowed_tiers: List[models.TierEnum])` inspects `user.tier`:

- **`Free`**: Unlocks basic mock tests, question practice, dashboard overview.
- **`Pro`**: Unlocks advanced mock tests, detailed performance metrics, daily practice sheets (DPS).
- **`AI_Coach`**: Unlocks real-time AI Tutor streaming (`/api/ai-coach/chat/stream`), Exam Readiness & College Prediction (`/api/ai-coach/readiness/{user_id}`), and Daily AI Study Plans (`/api/ai-coach/study-plan/today`).

3. Exhaustive REST API Specification (With Headers, Payloads & Responses)

3.1 Authentication Subsystem

`POST /api/auth/register`

- **Purpose:** Create a new student account with bcrypt password hashing and default profile fields.
- **Access:** Public
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
- **Request Body Schema:**

```
{
  "name": "Aryan Sharma",
  "email": "aryan@example.com",
  "password": "Password123!"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

```
{
  "success": true,
  "message": "User registered successfully"
}
```

  - **`400 Bad Request`** :

```
{
  "detail": "Email already registered"
}
```

  - **`422 Unprocessable Entity`** : If required fields (`name` , `email` , `password`) are missing or malformed.

`POST /api/auth/login`

- **Purpose:** Authenticate credentials using bcrypt verification, support seeded credentials & demo bypass, and issue active user token.
- **Access:** Public
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
- **Request Body Schema:**

```
{
  "email": "demo@pragativeer.com",
  "password": "Pv9#kL2$mQ8!vX4@"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK (Student)`** :

```
{
  "success": true,
  "role": "user",
  "token": "b0e17378-864d-439e-8806-708dc48a891b"
}
```

  - **`200 OK (Admin)`** :

```
{
  "success": true,
  "role": "admin",
  "token": "admin"
}
```

  - **`401 Unauthorized`** :

```
{
  "detail": "Invalid email or password"
}
```

  - **`422 Unprocessable Entity`** : If email or password is omitted.

`POST /api/auth/otp-request`

- **Purpose:** Request one-time verification code for phone/email authentication.
- **Access:** Public
- **Request Headers:**
  - `Accept: application/json`
- **Request Query Parameters:**
  - `phone_or_email` (string, required): e.g. `9876543210` or `demo@pragativeer.com`
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

```
{
  "message": "Verification code dispatched successfully",
  "debug_otp": "123456"
}
```

  - **`422 Unprocessable Entity`** : If `phone_or_email` is missing.

`POST /api/auth/otp-verify`

- **Purpose:** Verify OTP code and return session token.
- **Access:** Public
- **Request Headers:**
  - `Accept: application/json`
- **Request Query Parameters:**
  - `phone_or_email` (string, required): Phone number or email
  - `code` (string, required): 6-digit verification code
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`**:

    ```
    {
      "status": "authenticated",
      "token": "mock_jwt_session_token_string"
    }
    ```

  - **`400 Bad Request`**: `{"detail": "Invalid or expired OTP code"}`

3.2 Users, Session Storage & Avatar Uploads

Single-Device Session Storage Architecture

PragatiVeer v2 enforces a strict single-device login policy:

1. Upon successful login (`/login`), the browser generates a random UUID: `const sessionToken = crypto.randomUUID()` .
2. The browser writes this token to `localStorage.setItem("active_session_token", sessionToken)` .
3. The browser makes an immediate `POST /api/users/{user_id}/session` request, storing `active_session_token` in the database.
4. On subsequent page loads, `useActiveUser` calls `GET /api/users/{user_id}` . If the database's `active_session_token` does not match the browser's `localStorage` session token, the student has logged in on another device, triggering an automatic logout and redirect.

`GET /api/users/{user_id}`

- **Purpose:** Retrieve student profile and automatically compute consecutive-day streaks based on `last_active` .
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student user UUID
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**

- **`200 OK`** :

```
{
  "id": "b0e17378-864d-439e-8806-708dc48a891b",
  "phone": "9999999999",
  "email": "demo@pragativeer.com",
  "name": "Rahul Sharma",
  "student_class": "12",
  "medium": "English",
  "target_exam": "JEE_Main",
  "school": "Delhi Public School",
  "city": "New Delhi",
  "state": "Delhi",
  "tier": "Al_Coach",
  "streak_count": 6,
  "longest_streak": 6,
  "profile_image_url": "http://localhost:9000/avatars/user_avatar.jpg",
  "active_session_token": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "last_active": "2026-09-04T06:00:00.000000",
  "created_at": "2026-08-01T10:00:00.000000"
}
```

- **`404 Not Found`** : `{"detail": "User not found"}`
- **`422 Unprocessable Entity`** : If `user_id` is not a valid UUID format.

`PUT /api/users/{user_id}`

- **Purpose:** Update profile attributes (target exam, school, grade class, state).
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Request Body Schema:**

```
{
  "name": "Rahul Sharma",
  "target_exam": "JEE_Advanced",
  "student_class": "12",
  "school": "DPS RK Puram",
  "state": "Delhi"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Returns updated `models.User` object.
  - **`404 Not Found`** : `{"detail": "User not found"}`

- **`422 Unprocessable Entity`** : Malformed JSON or invalid UUID.

`POST /api/users/{user_id}/session`

- **Purpose:** Update single active session token in database to prevent multi-device concurrent logins.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Request Body Schema:**

```
{
  "session_token": "4f3b2a1c-9e8d-7c6b-5a4f-3e2d1c0b9a8f"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Returns `models.User` with updated `active_session_token` .
  - **`404 Not Found`** : `{"detail": "User not found"}`
  - **`422 Unprocessable Entity`** : Invalid UUID or missing `session_token` .

`POST /api/users/{user_id}/avatar`

- **Purpose:** Multipart form image upload for student avatar. Uploads image to MinIO S3 bucket `avatars` and updates `user.profile_image_url` .
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: multipart/form-data; boundary=----WebKitFormBoundary...`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Request Form Data:**
  - `file` : Image binary (`image/jpeg` , `image/png` , `image/webp`)
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

```
{
  "success": true,
  "profile_image_url": "http://localhost:9000/avatars/b0e17378_avatar.jpg"
}
```

- **`400 Bad Request`** : If uploaded file is not a supported image format.
- **`404 Not Found`** : `{"detail": "User not found"}`

`GET /api/users/{user_id}/attempts`

- **Purpose:** List historical exam test attempts for a student, with scores, accuracy, and timestamps.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Array of `models.Attempt` records:

```
[
  {
    "id": "e4f5a6b7-c8d9-0e1f-2a3b-4c5d6e7f8a9b",
    "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
    "source_type": "Test",
    "source_id": "b6faa295-3bf5-4de2-a587-d72b889629c4",
    "score": 184,
    "max_score": 300,
    "accuracy": 74.2,
    "time_taken": 9840,
    "retake_count": 0,
    "created_at": "2026-09-02T14:30:00.000000"
  }
]
```

  - **`404 Not Found`** : If student does not exist.

3.3 CBT Testing Engine & Question Bank

`GET /api/tests`

- **Purpose:** List active examination tests (Mock Tests, Weekly Tests, PYQs, DPS).
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`

- `Cookie: auth_token=<user_id>`

- **Request Query Parameters:**
  - `exam_pattern` (string, optional): `JEE_Main` | `JEE_Advanced` | `NEET`
  - `type` (string, optional): `Mock` | `Weekly` | `PYQ` | `DPS`. If omitted, automatically excludes `DPS` to keep Mock Tests separate.
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`**: Array of `models.Test` objects:

```
[
  {
    "id": "b6faa295-3bf5-4de2-a587-d72b889629c4",
    "title": "JEE Main Full Mock Test #1 (2025)",
    "type": "Mock",
    "exam_pattern": "JEE_Main",
    "sections": {
      "Physics Section A": { "questions": ["uuid-1", "uuid-2"], "time_limit": 3600 },
      "Chemistry Section A": { "questions": ["uuid-3", "uuid-4"], "time_limit": 3600 }
    },
    "total_time": 10800,
    "difficulty": "Mixed",
    "is_active": true
  }
]
```

`GET /api/tests/{test_id}`

- **Purpose:** Retrieve specific test configuration and section mapping.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `test_id` (UUID, required): Test ID
- **Request Query Parameters:**
  - `user_id` (UUID, optional): Optional student UUID to check eligibility
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`**: Returns single `models.Test` record.
  - **`404 Not Found`**: `{"detail": "Requested examination structure not found."}`
  - **`422 Unprocessable Entity`**: Invalid UUID format.

`GET /api/tests/{test_id}/questions`

- **Purpose:** Fetch all full Question records belonging to a test's sections.

- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `test_id` (UUID, required): Test ID
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Array of `models.Question` entities:

```
[
  {
    "id": "00000000-0000-0000-0000-000000000001",
    "subject": "Physics",
    "topic": "Kinematics",
    "subtopic": "Relative Motion",
    "difficulty": "Medium",
    "type": "MCQ_Single",
    "question_text": "A particle moves along a straight line with constant acceleration...",
    "question_text_hi": "एक कण स्थिर त्वरण के साथ एक सीधी रेखा में गति करता है...",
    "options": {
      "A": "10 m/s",
      "B": "20 m/s",
      "C": "30 m/s",
      "D": "40 m/s"
    },
    "correct_answer": { "correct": "B" },
    "solution_text": "Using kinematic formula v = u + at...",
    "language": "Bilingual",
    "exam": "JEE_Main",
    "is_active": true
  }
]
```

  - **`404 Not Found`** : If `test_id` is invalid or contains no questions.

`GET /api/questions/{question_id}`

- **Purpose:** Fetch single question data with complete step-by-step solution text.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `question_id` (UUID, required)
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**

- **`200 OK`** : Single `models.Question` record.
- **`404 Not Found`** : `{"detail": "Requested question not found."}`

`POST /api/attempts/submit`

- **Purpose:** Core grading engine. Auto-grades student answers, calculates accuracy %, tracks retakes, and automatically records mistakes into `models.Mistake` with SuperMemo-2 spaced repetition intervals.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Body Schema:**

```
{
  "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
  "source_type": "Test",
  "source_id": "b6faa295-3bf5-4de2-a587-d72b889629c4",
  "time_taken": 3600,
  "answers": [
    {
      "question_id": "00000000-0000-0000-0000-000000000001",
      "selected_answer": "B",
      "is_subjective": false,
      "time_spent": 45
    }
  ]
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Returns graded `models.Attempt` record with total score, accuracy %, and mistake IDs recorded.
  - **`404 Not Found`** : If `user_id` or test source does not exist.
  - **`422 Unprocessable Entity`** : Malformed payload.

3.4 AI Coach, Tutor Stream & Spaced Repetition

`POST /api/ai-coach/chat/stream`

- **Purpose:** Real-time conversational AI tutor for instant doubt resolution. Streams response tokens using Server-Sent Events (SSE).
- **Access:** Gated to `Al_Coach` tier (`HTTP 403` if Free or Pro)
- **Request Headers:**
  - `Accept: text/event-stream`
  - `Content-Type: application/json`
  - `Cache-Control: no-cache`

    - `Cookie: auth_token=<user_id>`
- **Request Body Schema:**

```
{
  "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
  "message": "Why does friction act forward on the driven wheel of an accelerating car?"
}
```

- **Response Headers:**
    - `Content-Type: text/event-stream; charset=utf-8`
    - `Cache-Control: no-cache, no-transform`
    - `Connection: keep-alive`
    - `X-Accel-Buffering: no`
- **Response Stream Chunks:**

```
data: When
data:  a
data:  car
data:  accelerates,
data:  the
data:  engine
data:  applies
data:  torque...
```

- **Status Codes:**
    - **`200 OK`** : Server-Sent Events stream initiated.
    - **`403 Forbidden`** : `{"detail": "Access denied. Gated to AI_COACH tier."}`
    - **`404 Not Found`** : `{"detail": "Student profile not found."}`
    - **`422 Unprocessable Entity`** : If `user_id` or `message` is missing.

`GET /api/ai-coach/study-plan/today`

- **Purpose:** Retrieve today's personalized AI study tasks generated nightly by Celery Beat scheduler based on weak areas and pending mistakes.
- **Access:** Gated to `Al_Coach` tier (`HTTP 403` if Free or Pro)
- **Request Headers:**
    - `Accept: application/json`
    - `Cookie: auth_token=<user_id>`
- **Request Query Parameters:**
    - `user_id` (UUID, required): Student UUID
- **Response Headers:**
    - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
    - **`200 OK`** :

```
{
  "id": "7f8e9d0a-1b2c-3d4e-5f6a-7b8c9d0e1f2a",
  "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
  "date": "2026-09-04",
  "tasks": [
    {
      "type": "Revision",
      "description": "Revise Rotational Mechanics",
      "duration_minutes": 15,
      "status": "Pending"
    },
    {
      "type": "Mistake_Review",
      "description": "Daily Mistake Review Session",
      "duration_minutes": 10,
      "status": "Pending"
    },
    {
      "type": "DPS",
      "description": "Attempt Rotational Mechanics Practice Sheet",
      "duration_minutes": 20,
      "status": "Pending"
    }
  ],
  "is_completed": false
}
```


- **`403 Forbidden`** : If user is not on `Al_Coach` tier.
- **`404 Not Found`** : `{"detail": "No study plan generated for today yet."}`
- **`422 Unprocessable Entity`** : If `user_id` query param is omitted.

`GET /api/ai-coach/readiness/{user_id}`

- **Purpose:** Computes weighted exam readiness score, predicts All India Rank (AIR), and estimates probability of admission to top IITs/AIIMS colleges.
- **Access:** Gated to `Al_Coach` tier
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

```
{
  "readiness_score": 78.4,
  "predicted_rank": 2840,
  "target_exam": "JEE_Main",
  "predicted_college": "IIT Delhi (Mechanical / Chemical)",
  "probability": 84.5,
  "weak_topics": ["Thermodynamics", "Wave Optics"],
  "strong_topics": ["Kinematics", "Electrostatics"]
}
```


- **`403 Forbidden`** : If user is not subscribed to `AI_Coach` .
- **`404 Not Found`** : If user does not exist.

`GET /api/mistakes/today`

- **Purpose:** Retrieve mistake notebook items scheduled for review today based on SuperMemo-2 (SM-2) spaced repetition interval.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Query Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Array of `models.Mistake` items with associated question context:


```
[
  {
    "id": "1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d",
    "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
    "question_id": "00000000-0000-0000-0000-000000000001",
    "mistake_type": "Calculation_Error",
    "review_count": 1,
    "interval_days": 3,
    "ease_factor": 2.5,
    "next_review": "2026-09-04",
    "resolved": false
  }
]
```


  - **`422 Unprocessable Entity`** : If `user_id` parameter is missing.

`POST /api/mistakes/review`

- **Purpose:** Update spaced repetition schedule after reviewing a mistake item (SM-2 algorithm calculates next review interval based on student recall quality 0-5).
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Body Schema:**

```
{
  "mistake_id": "1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d",
  "quality": 4
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`**：

```
{
  "success": true,
  "next_review": "2026-09-10",
  "interval_days": 6,
  "ease_factor": 2.6
}
```

  - **`404 Not Found`**：`{"detail": "Mistake record not found"}`

3.5 Subscription, Billing & Gateways

`POST /api/billing/create-order`

- **Purpose:** Initiates Razorpay payment order for tier upgrade (`Pro` or `Al_Coach`).
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Body Schema:**

```
{
  "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
  "tier": "Al_Coach",
  "amount": 299900,
  "coupon_code": "DISCOUNT20"
}
```

- **Response Headers:**

    - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
    - **`200 OK`**:

```
{
  "order_id": "order_EKf34dhu72ndD2",
  "amount": 239920,
  "currency": "INR",
  "key_id": "rzp_test_YOUR_KEY"
}
```

    - **`400 Bad Request`**: `{"detail": "Invalid tier or invalid coupon code."}`

`POST /api/billing/verify-payment`

- **Purpose:** Verifies Razorpay payment signature HMAC-SHA256, records transaction, and upgrades user's tier.
- **Access:** Authenticated User
- **Request Headers:**
    - `Accept: application/json`
    - `Content-Type: application/json`
    - `Cookie: auth_token=<user_id>`
- **Request Body Schema:**

```
{
  "user_id": "b0e17378-864d-439e-8806-708dc48a891b",
  "razorpay_order_id": "order_EKf34dhu72ndD2",
  "razorpay_payment_id": "pay_29QQoUBi66xm2f",
  "razorpay_signature": "9ef4705b83f6c19ff76543210abcd1234567890abcdef1234567890abcdef",
  "tier": "Al_Coach"
}
```

- **Response Headers:**
    - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
    - **`200 OK`**:

```
{
  "success": true,
  "tier": "Al_Coach",
  "message": "Payment verified and tier upgraded successfully."
}
```

    - **`400 Bad Request`**: `{"detail": "Payment verification failed. Invalid signature."}`

`POST /api/billing/webhook/razorpay`

- **Purpose:** Asynchronous payment confirmation webhook from Razorpay for recurring charges or asynchronous UPI captures.

- **Access:** Public (Protected by HMAC Webhook Secret)
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `X-Razorpay-Signature: <hmac_sha256_hex_digest>`
- **Request Body Schema:** Standard Razorpay Webhook Event payload:

```
{
  "entity": "event",
  "event": "payment.captured",
  "payload": {
    "payment": {
      "entity": {
        "id": "pay_29QQoUBi66xm2f",
        "amount": 239920,
        "status": "captured",
        "order_id": "order_EKf34dhu72ndD2"
      }
    }
  }
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : `{"status": "ok"}`
  - **`400 Bad Request`** : `{"detail": "Invalid webhook signature."}`

3.6 Admin Control Panel Subsystem

All Admin endpoints require superadmin access:

- **Required Cookie:** `Cookie: auth_token=admin`
- **Default Superadmin:** `admin@pragativeer.com` / `admin123`

`GET /api/admin/settings`

- **Purpose:** Retrieve system-wide runtime configuration (maintenance mode, free trial flags, AI model selection).
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

```
{
  "maintenance_mode": false,
  "allow_free_trials": true,
  "ai_model": "gemini-2.5-pro",
  "max_daily_questions": 150
}
```

- **`401 Unauthorized`** : If admin cookie is missing or invalid.

`PUT /api/admin/settings`

- **Purpose:** Update system-wide configuration settings.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=admin`
- **Request Body Schema:**

```
{
  "maintenance_mode": false,
  "allow_free_trials": true,
  "ai_model": "gemini-2.5-pro"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Returns updated settings object.
  - **`401 Unauthorized`** : If unauthorized.

`GET /api/admin/tests`

- **Purpose:** List all tests, PYQs, and DPS (including drafts and inactive tests).
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Array of all `models.Test` records.

`POST /api/admin/tests/upload`

- **Purpose:** Upload PDF exam paper to MinIO storage and dispatch background AI parsing task to Celery worker.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: multipart/form-data; boundary=----WebKitFormBoundary...`
  - `Cookie: auth_token=admin`
- **Request Form Data:**
  - `file` : PDF binary (`application/pdf`)
  - `exam_pattern` : `JEE_Main` | `JEE_Advanced` | `NEET`
  - `title` : Test name string
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

    ```
    {
      "task_id": "c56d78ef-12ab-34cd-56ef-7890abcdef12",
      "status": "Processing",
      "message": "PDF uploaded to MinIO. Celery AI parser dispatched."
    }
    ```

  - **`400 Bad Request`** : If file is not a valid PDF.

`POST /api/admin/tests/create_manual`

- **Purpose:** Manually create a CBT mock test structure and map questions.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=admin`
- **Request Body Schema:**

```
{
  "title": "All India Grand Mock Test #5",
  "type": "Mock",
  "exam_pattern": "JEE_Main",
  "total_time": 10800,
  "difficulty": "Hard",
  "sections": {
    "Physics": { "questions": [], "time_limit": 3600 }
  }
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**

- **200 OK** : Created `models.Test` record.

`PUT /api/admin/tests/{test_id}/toggle`

- **Purpose:** Toggle test `is_active` status (publish/unpublish).
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Request Path Parameters:**
  - `test_id` (UUID, required)
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **200 OK** : `{"id": "...", "is_active": true}`
  - **404 Not Found** : If test does not exist.

`DELETE /api/admin/tests/{test_id}`

- **Purpose:** Permanently delete a test and dissociate question mappings.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Request Path Parameters:**
  - `test_id` (UUID, required)
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **200 OK** : `{"success": true, "message": "Test deleted successfully."}`
  - **404 Not Found** : If test does not exist.

`GET /api/admin/coupons`

- **Purpose:** List all promotional discount coupons.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **200 OK** : Array of `models.Coupon` records.

`POST /api/admin/coupons`

- **Purpose:** Create a new promotional discount coupon.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
  - `Cookie: auth_token=admin`
- **Request Body Schema:**

```
{
  "code": "FESTIVE25",
  "discount_percentage": 25,
  "max_uses": 500,
  "valid_until": "2026-12-31T23:59:59.000000"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : Returns created `models.Coupon` entity.

`DELETE /api/admin/coupons/{coupon_id}`

- **Purpose:** Delete promotional coupon.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Request Path Parameters:**
  - `coupon_id` (UUID, required)
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** : `{"success": true, "message": "Coupon deleted"}`
  - **`404 Not Found`** : `{"detail": "Coupon not found"}`

`GET /api/admin/students`

- **Purpose:** List all registered students with attempts and accuracy metrics.
- **Access:** Admin
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=admin`
- **Response Headers:**

    - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
    - **`200 OK`** : Array of student summary objects.

`GET /api/admin/students/{user_id}`

- **Purpose:** Detailed student profile with weak topics, recent tests, transactions.
- **Access:** Admin
- **Request Headers:**
    - `Accept: application/json`
    - `Cookie: auth_token=admin`
- **Request Path Parameters:**
    - `user_id` (UUID, required)
- **Response Headers:**
    - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
    - **`200 OK`** : Detailed student dossier.
    - **`404 Not Found`** : Student not found.

`PATCH /api/admin/students/{user_id}`

- **Purpose:** Modify student tier, streak count, school, city, or exam.
- **Access:** Admin
- **Request Headers:**
    - `Accept: application/json`
    - `Content-Type: application/json`
    - `Cookie: auth_token=admin`
- **Request Path Parameters:**
    - `user_id` (UUID, required)
- **Request Body Schema:**

```
{
  "tier": "Al_Coach",
  "streak_count": 15
}
```

- **Response Headers:**
    - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
    - **`200 OK`** : Returns modified `models.User` record.
    - **`404 Not Found`** : Student not found.

3.7 Parent Subsystem & Viral Growth

`POST /api/parent/register`

- **Purpose:** Register parent user profile in `models.Parent`.
- **Access:** Public
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
- **Request Body Schema:**

```
{
  "name": "Rajesh Kumar",
  "phone": "9876543210",
  "email": "parent@example.com",
  "password_hash": "hashed_password"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`**:

```
{
  "id": "c1d2e3f4-5a6b-7c8d-9e0f-1a2b3c4d5e6f",
  "name": "Rajesh Kumar",
  "phone": "9876543210",
  "email": "parent@example.com"
}
```

  - **`400 Bad Request`**: If phone or email is already registered.

`POST /api/parent/link-student`

- **Purpose:** Link a student's profile to a parent account using student's phone or email.
- **Access:** Parent User
- **Request Headers:**
  - `Accept: application/json`
- **Request Query Parameters:**
  - `parent_id` (UUID, required): Parent UUID
  - `student_phone_or_email` (string, required): Student's registered phone or email
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`**:

```
{
  "success": true,
  "message": "Student linked to parent account successfully."
}
```

- **404 Not Found** : `{"detail": "Student or parent account not found."}`

GET /api/parent/{parent_id}/dashboard/{student_id}

- **Purpose:** Generate complete parent report with test summary, syllabus progress, and AI recommendations.
- **Access:** Parent User
- **Request Headers:**
  - `Accept: application/json`
- **Request Path Parameters:**
  - `parent_id` (UUID, required)
  - `student_id` (UUID, required)
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **200 OK** :

```
{
  "student_name": "Rahul Sharma",
  "target_exam": "JEE_Main",
  "tests_attempted": 8,
  "average_accuracy": 74.2,
  "current_streak": 6,
  "ai_insight": "Rahul is performing exceptionally well in Kinematics but needs focused revision on Thermodynam
}
```

  - **404 Not Found** : If parent or student record does not exist.

GET /api/growth/referral/{user_id}

- **Purpose:** Generates referral code, viral share URL, referral count, and milestone unlocks.
- **Access:** Authenticated User
- **Request Headers:**
  - `Accept: application/json`
  - `Cookie: auth_token=<user_id>`
- **Request Path Parameters:**
  - `user_id` (UUID, required): Student UUID
- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **200 OK** :

```
{
  "referral_code": "PV-RAHUL-1A2B",
  "referral_url": "http://localhost:3000/signup?ref=PV-RAHUL-1A2B",
  "total_referred": 3,
  "unlocked_rewards": ["1 Free Mock Test", "7 Days AI Coach Trial"]
}
```

- **`404 Not Found`** : If student does not exist.

3.8 Next.js Route Handlers & Edge Handlers

`POST /api/auth/forgot-password`

- **Location:** `src/app/api/auth/forgot-password/route.ts`
- **Runtime:** Node.js App Router
- **Access:** Public
- **Request Headers:**
  - `Accept: application/json`
  - `Content-Type: application/json`
- **Request Body Schema:**

```
{
  "email": "demo@pragativeer.com"
}
```

- **Response Headers:**
  - `Content-Type: application/json`
- **Status Codes & Response Payloads:**
  - **`200 OK`** :

```
{
  "success": true,
  "message": "Reset instructions sent."
}
```

  - **`400 Bad Request`** : `{"detail": "Email address is required."}`

`GET /api/og/progress`

- **Location:** `src/app/api/og/progress/route.tsx`
- **Runtime:** Edge (`@vercel/og` ImageResponse)
- **Access:** Public
- **Request Headers:**
  - `Accept: image/png, image/*`
- **Request Query Parameters:**
  - `streak` (integer, optional): Day streak count (e.g. `12`)
  - `accuracy` (number, optional): Accuracy percentage (e.g. `78`)

    - `name` (string, optional): Student name (e.g. `Rahul`)
- **Response Headers:**
    - `Content-Type: image/png`
    - `Cache-Control: public, max-age=31536000, immutable`
- **Status Codes & Response Payloads:**
    - **`200 OK`**: Dynamic 1200x630 PNG social card image.

4. Planned APIs (Roadmap / Frontend Mock Status)

The following endpoints are specified in [PRD.md](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/PRD.md) or rendered using client-side mock data in the UI:

Planned API	Expected Route	Frontend UI File	Backend Provisioning Needed
Leaderboard API	`GET /api/leaderboard`	[leaderboard/page.tsx](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/src/app/dashboard/leaderboard/page.tsx)	Redis sorted set queries (`ZREVRANGE`) by exam/state
Student Analytics API	`GET /api/users/{id}/analytics`	[analytics/page.tsx](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/src/app/dashboard/analytics/page.tsx)	Weekly score trajectories and topic heatmap accuracies
Streak Rewards API	`GET /api/streaks/rewards`	[streaks/page.tsx](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/src/app/dashboard/streaks/page.tsx)	Gamification milestone reward unlocks
Syllabus Progress Tracker	`GET/PUT /api/syllabus/progress`	[progress/page.tsx](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/src/app/dashboard/progress/page.tsx)	Chapter-by-chapter completion % per subject
Community Doubts Feed	`GET/POST /api/community/doubts`	[community/page.tsx](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/src/app/dashboard/community/page.tsx)	Discussion forum posts, replies, and upvotes
Study Materials Library	`GET /api/materials`	[materials/page.tsx](file:///c:/Users/naman/OneDrive/Desktop/pragativeer-v2/src/app/dashboard/materials/page.tsx)	MinIO PDF formula sheet catalog and access controls

Planned API	Expected Route	Frontend UI File	Backend Provisioning Needed
Voice AI & Live Battle	`/ws/battles` , `/api/ai-coach/voice`	Future Scope	WebSockets for multiplayer CBT battles & Whisper voice

5. Seeded Credentials Reference

The platform provides two pre-configured accounts for administration and end-to-end evaluation:

Role	Email	Password	Tier / Features
Super Admin	`admin@pragativeer.com`	`admin123`	Full access to `/admin` management console & question bank
AI Coach Student	`demo@pragativeer.com`	`Pv9#kL2$mQ8!vX4@`	Full AI Coach access (AI Tutor SSE stream, College Predictor, CBT Mock Engine, Spaced Repetition)