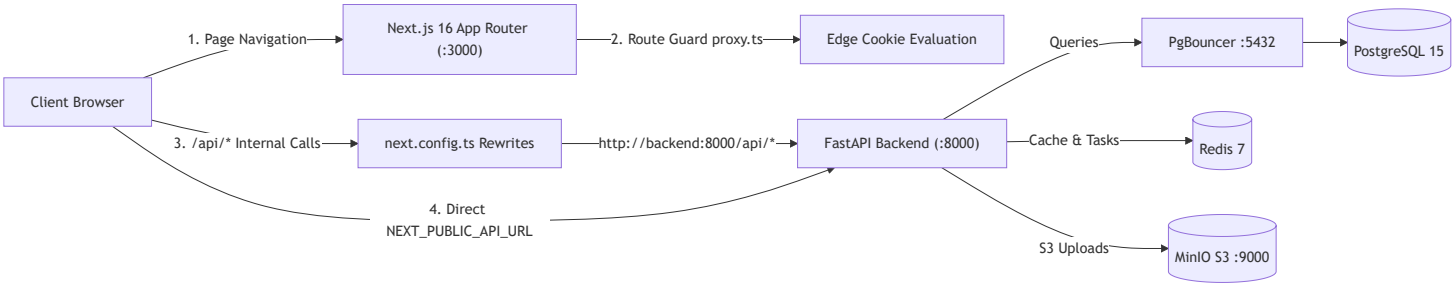
PragatiVeer v2 — Pagewise UI & API Specification Map

> **Document Type:** Comprehensive Pagewise Architecture, Button-to-API Interaction, and UI Specification Map
> **Target Audience:** Frontend Engineers, Full-Stack Developers, Product Managers, QA & DevOps
> **Last Verified:** September 2026
> **Frontend Engine:** Next.js 16 (App Router, React 19, TypeScript, TailwindCSS, Zustand)
> **Backend Engine:** FastAPI (Python 3.11) + PostgreSQL 15 (PgBouncer) + Redis 7 + MinIO S3

1. Architectural & Network Request Flow

1.1 Client-to-Backend Network Pipeline

PragatiVeer v2 operates two primary communication paths between the Next.js frontend and the FastAPI backend:



1. **Proxy Middleware (`src/proxy.ts`):** Evaluates `auth_token` cookie on all `/dashboard/*` and `/admin/*` routes.
   - Non-authenticated visitors heading to `/dashboard` are redirected to `/login` .
   - Non-admin visitors (`token !== 'admin'`) heading to `/admin` are redirected to `/login` .
2. **Reverse Proxy Rewrites (`next.config.ts`):** Inside Docker, `/api/:path*` is automatically rewritten to `http://backend:8000/api/:path*` .
3. **Direct Client Fetching:** Certain client stores (e.g. `cbtStore.ts` and `src/app/cbt/[examId]/page.tsx`) use `NEXT_PUBLIC_API_URL` (default `http://localhost:8000`) for high-throughput examination requests.
4. **Session Token Enforcement:** `localStorage.getItem("active_session_token")` provides single-device session validation against `/api/users/{id}/session` .

2. Master Page Directory

Route URL	Component File Location	Access Level	Primary E
`/`	`src/app/page.tsx`	Public	None (Sta
`/about`	`src/app/about/page.tsx`	Public	None (Info
`/contact`	`src/app/contact/page.tsx`	Public	None (Sta
`/login`	`src/app/login/page.tsx`	Public	`POST /api` `POST /api`
`/signup`	`src/app/signup/page.tsx`	Public	`POST /api` `POST /api`
`/forgot-password`	`src/app/forgot-password/page.tsx`	Public	`POST /api`
`/upgrade`	`src/app/upgrade/page.tsx`	Public / Auth	None (Ro
`/checkout`	`src/app/checkout/page.tsx`	Authenticated	`POST /api` `POST /api`

Route URL	Component File Location	Access Level	Primary E
/cbt/[examId]	src/app/cbt/[examId]/page.tsx	Authenticated	GET /api/ GET /api/ POST /api
/dashboard	src/app/dashboard/page.tsx	Student (Free +)	GET /api/ GET /api/ GET /api/
/dashboard/cbt	src/app/dashboard/cbt/page.tsx	Student (Free +)	GET /api/ GET /api/
/dashboard/cbt/history/[attemptId]	src/app/dashboard/cbt/history/[attemptId]/page.tsx	Student (Free +)	GET /api/ GET /api/
/dashboard/dpp	src/app/dashboard/dpp/page.tsx	Student (Free +)	GET /api/ GET /api/
/dpp/review/[dppId]	src/app/dpp/review/[dppId]/page.tsx	Student (Free +)	Local DPP
/dashboard/analytics	src/app/dashboard/analytics/page.tsx	Student (Free +)	GET /api/ accuracies
/dashboard/progress	src/app/dashboard/progress/page.tsx	Student (Free +)	GET /api/
/dashboard/streaks	src/app/dashboard/streaks/page.tsx	Student (Free +)	GET /api/ tokens)
/dashboard/leaderboard	src/app/dashboard/leaderboard/page.tsx	Student (Free +)	Client rank
/dashboard/materials	src/app/dashboard/materials/page.tsx	Student (Free +)	Formula s
/dashboard/rank-predictor	src/app/dashboard/rank-predictor/page.tsx	Student (Free +)	Percentile
/dashboard/community	src/app/dashboard/community/page.tsx	Student (Free +)	Student d
/dashboard/settings	src/app/dashboard/settings/page.tsx	Student (Free +)	GET /api/ POST /api
/dashboard/ai-coach	src/app/dashboard/ai-coach/page.tsx	Student (AI_COACH)	GET /api/ GET /api/
/dashboard/ai-coach/tutor	src/app/dashboard/ai-coach/tutor/page.tsx	Student (AI_COACH)	POST /api
/dashboard/ai-coach/study-plan	src/app/dashboard/ai-coach/study-plan/page.tsx	Student (AI_COACH)	GET /api/
/dashboard/ai-coach/readiness	src/app/dashboard/ai-coach/readiness/page.tsx	Student (AI_COACH)	GET /api/
/dashboard/ai-coach/mistakes	src/app/dashboard/ai-coach/mistakes/page.tsx	Student (AI_COACH)	GET /api/ POST /api
/dashboard/ai-coach/cbt-dpp	src/app/dashboard/ai-coach/cbt-dpp/page.tsx	Student (AI_COACH)	GET /api/ GET /api/
/dashboard/ai-coach/cbt-dpp/[id]	src/app/dashboard/ai-coach/cbt-dpp/[id]/page.tsx	Student (AI_COACH)	Submits to

Route URL	Component File Location	Access Level	Primary E
`/dashboard/ai-coach/cbt-dpp/history/[attemptId]`	`src/app/dashboard/ai-coach/cbt-dpp/history/[attemptId]/page.tsx`	Student (`AI_COACH`)	`GET /api/` `GET /api/`
`/dashboard/ai-coach/heatmap`	`src/app/dashboard/ai-coach/heatmap/page.tsx`	Student (`AI_COACH`)	Mistake fr
`/dashboard/ai-coach/college-predictor`	`src/app/dashboard/ai-coach/college-predictor/page.tsx`	Student (`AI_COACH`)	`GET /api/` cutoff map
`/dashboard/ai-coach/time-optimization`	`src/app/dashboard/ai-coach/time-optimization/page.tsx`	Student (`AI_COACH`)	Time-per-
`/dashboard/ai-coach/strategy`	`src/app/dashboard/ai-coach/strategy/page.tsx`	Student (`AI_COACH`)	Exam tem
`/dashboard/ai-coach/weekly-progress`	`src/app/dashboard/ai-coach/weekly-progress/page.tsx`	Student (`AI_COACH`)	Weekly AI
`/admin`	`src/app/admin/page.tsx`	Admin	Real-time
`/admin/students`	`src/app/admin/students/page.tsx`	Admin	`GET /api/`
`/admin/students/[studentId]`	`src/app/admin/students/[studentId]/page.tsx`	Admin	`GET /api/` `PATCH /ap`
`/admin/tests`	`src/app/admin/tests/page.tsx`	Admin	`GET /api/` `PUT /api/` `DELETE /a` `POST /api`
`/admin/questions`	`src/app/admin/questions/page.tsx`	Admin	Global qu
`/admin/coupons`	`src/app/admin/coupons/page.tsx`	Admin	`GET /api/` `POST /api` `DELETE /a`
`/admin/settings`	`src/app/admin/settings/page.tsx`	Admin	`GET /api/` `POST /api`
`/admin/agents`	`src/app/admin/agents/page.tsx`	Admin	AI Agent c
Policy Pages	`/privacy` , `/terms` , `/refund` , `/cancellation` , `/disclaimer`	Public	Static lega

3. Exhaustive Pagewise UI & API Breakdowns

3.1 Authentication & Onboarding Pages

A. Login Page (`/login`)

- **File Location:** `src/app/login/page.tsx`
- **Access Guard:** Public (Redirects authenticated users to `/dashboard` or `/admin`)
- **UI Components:**
  - Brand Logo linking to `/`
  - Email `<input type="email" />`
  - Password `<input type="password" />` with show/hide password toggle button
  - "Forgot password?" link to `/forgot-password`
  - "Sign In" submit button

- "Sign up" link to `/signup`

- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Show/Hide Password`	Toggles password visibility	`setShowPassword(!showPassword)`
`Sign In Button`	Submits form, authenticates user	`handleLogin(e)`
`Forgot password?`	Navigates to password reset	`/forgot-password`
`Sign up`	Navigates to registration	`/signup`

- **APIs Invoked:**
  i. `POST /api/auth/login`
    - **Trigger:** On form submit (`handleLogin`)
    - **Request Body:** `{ "email": "...", "password": "..." }`
    - **Response:** `{ "success": true, "role": "user" | "admin", "token": "UUID" | "admin" }`
    - **Side Effects:**
      - Sets `document.cookie = "auth_token=...; path=/; max-age=2592000; SameSite=Lax"`
      - Writes `localStorage.setItem("activeUserId", data.token)`
      - Generates `crypto.randomUUID()` session token and writes `localStorage.setItem("active_session_token", sessionToken)`
  ii. `POST /api/users/{userId}/session`
    - **Trigger:** Immediately following successful student login
    - **Request Body:** `{ "session_token": "<random_uuid>" }`
    - **Response:** `{ "status": "session_registered", ... }`
    - **Redirect:** `window.location.href = "/admin"` (if admin) or `"/dashboard"` (if student)

B. Signup Page (`/signup`)

- **File Location:** `src/app/signup/page.tsx`
- **Access Guard:** Public
- **UI Components:**
  - Name `<input type="text" />`
  - Email `<input type="email" />`
  - Password `<input type="password" />`
  - "Create Account" submit button
  - "Sign In" link to `/login`
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Create Account Button`	Submits registration	`handleSignup(e)`
`Sign in Link`	Navigates to login	`/login`

- **APIs Invoked:**
  i. `POST /api/auth/register`
    - **Trigger:** On form submit
    - **Request Body:** `{ "name": "...", "email": "...", "password": "..." }`
    - **Response:** `{ "success": true, "message": "User registered successfully" }`
  ii. `POST /api/auth/login` (Auto-login chaining)
    - **Trigger:** Immediately upon successful registration
    - **Request Body:** `{ "email": "...", "password": "..." }`
    - **Response:** `{ "success": true, "token": "<userId>", "role": "user" }`
  iii. `POST /api/users/{userId}/session`
    - Registers active session token and redirects to `/dashboard` .

C. Forgot Password Page (`/forgot-password`)

- **File Location:** `src/app/forgot-password/page.tsx`
- **Access Guard:** Public
- **UI Components:**
  - Email `<input type="email" />`
  - "Send Reset Link" button
  - Success message banner
  - "Back to login" link
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Send Reset Link`	Dispatches password reset email	`handleReset(e)`
`Back to login`	Navigates to `/login`	`/login`

- **APIs Invoked:**
  i. `POST /api/auth/forgot-password`
    - **Trigger:** On form submit
    - **Request Body:** `{ "email": "student@example.com" }`
    - **Response:** `{ "message": "If this email is registered, a password reset link has been sent." }`

3.2 Subscription & Checkout Subsystem

A. Pricing & Upgrade Page (`/upgrade`)

- **File Location:** `src/app/upgrade/page.tsx`
- **Access Guard:** Public / Authenticated
- **UI Components:**
  - Plan selection cards:
    - **Free Plan:** Basic mocks, limited practice sheets (Default)
    - **Pro Plan (₹3,499/yr):** Unlimited Full-Syllabus CBT Mocks, Detailed Analytics, Daily DPS
    - **AI Coach Plan (₹11,999/yr):** Real-time AI Tutor, College Predictor, Spaced Repetition Notebook, Adaptive Daily Study Plans
  - Feature comparison matrix
  - FAQ accordion
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Choose Pro`	Navigates to checkout for Pro	`router.push('/checkout?tier=Pro')`
`Upgrade to AI Coach`	Navigates to checkout for AI Coach	`router.push('/checkout?tier=Al_Coach')`

B. Checkout & Payment Page (`/checkout`)

- **File Location:** `src/app/checkout/page.tsx`
- **Access Guard:** Authenticated (Redirects to `/login?redirect=/checkout?tier=...` if guest)
- **UI Components:**
  - Order summary sidebar (Plan name, subtotal, GST 18%, grand total)
  - Delivery/Student Address inputs (`Address` , `City` , `Pincode` , `Phone`)
  - Payment method selector: Razorpay / Cashfree / Stripe
  - "Proceed to Payment" button with active loader
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Proceed to Payment`	Initiates backend order & triggers Razorpay Modal	`handlePayment()`

- **APIs Invoked:**
  i. `POST /api/billing/create-order?user_id={userId}&plan_tier={tier}`
    - **Trigger:** On `Proceed to Payment` click
    - **Request Body:** `{ "gateway": "razorpay" }`
    - **Response:**

```
{
  "id": "order_XXXXXXXXXXXXXX",
  "amount": 1199900,
  "currency": "INR",
  "status": "created"
}
```

  ii. **Client-Side SDK:** Opens `window.Razorpay(options).open()`
  iii. `POST /api/billing/verify-payment`
    - **Trigger:** Razorpay `handler(response)` callback upon payment approval
    - **Request Body:**

```
{
  "gateway": "razorpay",
  "razorpay_order_id": "order_XXXXXX",
  "razorpay_payment_id": "pay_XXXXXX",
  "razorpay_signature": "signature_hash",
  "user_id": "<userId>",
  "tier": "Al_Coach"
}
```

    - **Response:** `{ "status": "success", "tier": "Al_Coach" }`
    - **Action on Success:** Updates local user tier and redirects to `/dashboard?payment=success`.

3.3 Computer-Based Testing (CBT) Examination Engine

A. Full-Screen CBT Examination Room (`/cbt/[examId]`)

- **File Location:** `src/app/cbt/[examId]/page.tsx`
- **Access Guard:** Authenticated (`Free` , `Pro` , or `AI_COACH` depending on test gating)
- **UI Components:**
  - Header: Candidate Name, Photo, Roll Number, Exam Title, Section Switcher (`Physics` , `Chemistry` , `Mathematics` or `Biology`)
  - Floating Exam Timer (Hours : Minutes : Seconds) with visual alert when under 5 minutes
  - Question Palette Matrix (Grid 1 to N with NTA standard colors: Green=Answered, Purple=Marked for Review, Red=Unanswered, Grey=Not Visited)
  - Question Panel: Question Text (KaTeX LaTeX rendering), Image Diagram from MinIO, Multiple Choice Options (`A` , `B` , `C` , `D`) or Numerical Input box
  - Action Bar:
    - "Save & Next" button
    - "Clear Response" button
    - "Mark for Review & Next" button
    - "Submit Test" button
  - Submit Confirmation Modal: Summary table of Answered, Unanswered, Marked for Review questions
- **Buttons & Interactive Elements:**

Button / Trigger	Action	State / Store Handler
`Option Selection (A/B/C/D)`	Selects radio answer	`saveAnswer(questionId, optionKey)`
`Numerical Input Change`	Enters integer/decimal answer	`saveAnswer(questionId, value)`
`Save & Next`	Commits answer and advances index	`handleNextQuestion()`
`Clear Response`	Resets current question answer	`clearAnswer(questionId)`

Button / Trigger	Action	State / Store Handler
`Mark for Review & Next`	Sets flagged review state and advances	`markForReview(questionId)`
`Question Palette Button [i]`	Direct jump to question index `i`	`setCurrentIndex(i)`
`Section Tab Switcher`	Filters question palette by Subject	`setActiveSection(sectionName)`
`Submit Test Button`	Opens final confirmation dialog	`setIsSubmitModalOpen(true)`
`Confirm Final Submission`	Dispatches submission payload	`submitAttemptToServer(userId)`

- **APIs Invoked:**
  i. `GET /api/tests/{examId}?user_id={userId}`
    - **Trigger:** On mount (`useEffect`)
    - **Response:** Test metadata, sections mapping, total duration, question ID sequence.
  ii. `GET /api/tests/{examId}/questions`
    - **Trigger:** On mount immediately following test metadata
    - **Response:** Array of full question objects (LaTeX text, options, subject, topic).
  iii. `POST /api/attempts/submit`
    - **Trigger:** On manual submission confirm OR auto-submit on countdown timeout
    - **Request Body (`src/store/cbtStore.ts`):**

```
{
  "test_id": "<examId>",
  "user_id": "<userId>",
  "time_spent": 10420,
  "answers": [
    {
      "question_id": "<qId>",
      "selected_option": "B",
      "time_spent": 115,
      "is_marked_for_review": false
    }
  ],
  "status": "Submitted"
}
```

    - **Response:**

```
{
  "id": "<attemptId>",
  "score": 240,
  "total_questions": 75,
  "accuracy": 82.5,
  "rank": 412
}
```

    - **Redirect:** Navigates directly to `/dashboard/cbt/history/{attemptId}` .

B. CBT Mock Directory Hub (`/dashboard/cbt`)

- **File Location:** `src/app/dashboard/cbt/page.tsx`
- **Access Guard:** Authenticated (`Free` +)
- **UI Components:**
  - Exam Filter Tabs: `All` , `JEE Main` , `JEE Advanced` , `NEET`
  - Active Full Mocks Grid (Test Card: Title, Pattern badge, Total Questions, Time limit, Attempts count, "Start Test" or "Review" button)
  - Recent Attempts History Table (Exam Title, Score, Accuracy, Date, "View Solution" button)
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Exam Filter Tabs`	Filters active tests by pattern	`setSelectedPattern(pattern)`
`Start Test Button`	Navigates into full-screen exam room	`router.push('/cbt/' + test.id)`
`View Solution Button`	Navigates to detailed attempt analysis	`router.push('/dashboard/cbt/history/' + attempt.id)`

- **APIs Invoked:**
  i. `GET /api/users/{userId}` (Fetches user's targeted exam e.g. `JEE_MAIN`)
  ii. `GET /api/tests?user_id={userId}&exam_pattern={pattern}` (Lists available mock tests)
  iii. `GET /api/users/{userId}/attempts` (Fetches historical attempt records)

C. Test Attempt Analysis & Solution Review (`/dashboard/cbt/history/[attemptId]`)

- **File Location:** `src/app/dashboard/cbt/history/[attemptId]/page.tsx`
- **Access Guard:** Authenticated
- **UI Components:**
  - Score summary card: Final Score, Total Marks, Accuracy %, Positive Marks, Negative Marks, Percentile
  - Question-by-question breakdown list:
    - Status badge: Correct (+4), Incorrect (-1), Unattempted (0)
    - Full LaTeX question rendering
    - Student's chosen answer vs Official Correct Answer
    - Step-by-step text solution
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Filter by Subject (All/Phys/Chem/Math)`	Filters questions	`setSelectedSubject(sub)`
`Filter by Status (Incorrect/Correct)`	Isolates student errors	`setStatusFilter(filter)`
`Back to Test Directory`	Navigates back	`/dashboard/cbt`

- **APIs Invoked:**
  i. `GET /api/users/{userId}/attempts` (Locates `attemptId`)
  ii. `GET /api/tests/{source_id}?user_id={userId}` (Fetches test structure)
  iii. `GET /api/questions/{questionId}` (Fetches detailed solutions per question)

D. Daily Practice Sheets (DPS / DPP) Test Center (`/dashboard/dpp`)

- **File Location:** `src/app/dashboard/dpp/page.tsx`
- **Access Guard:** Authenticated (`Free` + with tiered limits)
- **UI Components:**
  - Daily Sheets Calendar / Carousel:
    - Biology Ch4 DPS
    - Chemistry Ch4 DPS
    - Mathematics Ch4 DPS
    - Physics Ch4 DPS
  - Sheet Status: `Pending` | `Completed` (with Score & Accuracy)
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Solve DPS Button`	Launches CBT engine for that practice sheet	`router.push('/cbt/' + sheet.id)`
`Review Solution Button`	Opens attempt breakdown	`router.push('/dashboard/cbt/history/' + attempt.id)`

- **APIs Invoked:**

i. `GET /api/tests?type=DPS&exam_pattern={pattern}` (Fetches seeded DPS tests)
ii. `GET /api/users/{userId}/attempts` (Checks which DPS sheets have been completed)

3.4 Student Dashboard & Core Learning Hub

A. Main Student Dashboard (`/dashboard`)

- **File Location:** `src/app/dashboard/page.tsx`
- **Access Guard:** Authenticated (`Free` +)
- **UI Components:**
  - Header: Welcome message, Active Target Exam (`JEE Main` / `NEET`), Streak Badge
  - Performance Overview Grid: Tests Taken, Average Accuracy %, Study Streak, Target Score
  - Embedded Adaptive Study Plan Card (`AdaptiveStudyPlan.tsx`)
  - Embedded Mistake Heatmap Card (`MistakeHeatmap.tsx`)
  - Embedded Viral Referral Card (`ReferralCard.tsx`)
  - "Share Card" Button (Generates and downloads high-res progress PNG via `html-to-image`)
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Share Card Button`	Exports dashboard card as PNG	`handleExport()` (`html-to-image`)
`Start Today's DPS`	Launches daily practice	`/dashboard/dpp`
`Explore AI Coach`	Navigates to AI features	`/dashboard/ai-coach`

- **APIs Invoked:**
  i. `GET /api/users/{userId}` via `useActiveUser()`
  ii. `GET /api/ai-coach/study-plan/today?user_id={userId}` (via `AdaptiveStudyPlan`)
  iii. `GET /api/growth/referral/{userId}` (via `ReferralCard`)

B. Student Profile & Settings Page (`/dashboard/settings`)

- **File Location:** `src/app/dashboard/settings/page.tsx`
- **Access Guard:** Authenticated
- **UI Components:**
  - Avatar Profile Picture with MinIO upload button
  - Personal Information Form: First Name, Last Name, Email (read-only), Target Exam selector (`JEE Main` , `JEE Advanced` , `NEET`), School, State
  - "Save Changes" button
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Avatar File Upload Input`	Uploads image directly to MinIO	`handleAvatarUpload(e)`
`Save Changes Button`	Updates student profile	`handleSave()`

- **APIs Invoked:**
  i. `POST /api/users/{userId}/avatar`
    - **Trigger:** On file input change
    - **Request Form-Data:** `file: <binary_image>`
    - **Response:** `{ "url": "http://localhost:9000/avatars/user_avatar_....png" }`
  ii. `PUT /api/users/{userId}`
    - **Trigger:** On `Save Changes` click
    - **Request Body:**

```
{
  "name": "Rahul Sharma",
  "target_exam": "JEE_MAIN",
  "school": "Delhi Public School",
  "state": "Delhi"
}
```

- **Response:** Updated user record.

3.5 AI Coach Suite (`/dashboard/ai-coach/*`)

> **Access Restriction:** All pages in this section are gated to `tier == "AI_Coach"`. Users on `Free` or `Pro` tiers see an upgrade paywall with a direct link to `/checkout?tier=AI_Coach`.

A. AI Doubt Solver & Tutor Chat (`/dashboard/ai-coach/tutor`)

- **File Location:** `src/app/dashboard/ai-coach/tutor/page.tsx` & `src/components/AI/TutorChat.tsx`
- **Access Guard:** `AI_COACH` Tier
- **UI Components:**
  - Real-time chat message history (User messages in brand cyan, AI Tutor responses formatted in Markdown + KaTeX LaTeX equations)
  - Chat `<input />` with send button and typing indicator
  - Quick Suggestion Chips: *"Explain Photoelectric Effect"*, *"Solve JEE 2024 Math Integration"*, *"Give me a 3-step revision plan"*
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Send Message Button`	Submits doubt to streaming API	`sendMessage(text)`
`Suggestion Chips`	Populates input with prompt	`setInput(chipText)`

- **APIs Invoked:**
  i. `POST /api/ai-coach/chat/stream`
    - **Trigger:** On sending a message
    - **Request Headers:** `Content-Type: application/json`, `Accept: text/event-stream`
    - **Request Body:** `{ "user_id": "<userId>", "message": "Explain Lenz's Law" }`
    - **Response Stream:** `text/event-stream; charset=utf-8` (Server-Sent Events)
    - **Phase 2 Status:** Intentionally stubbed with friendly coming-soon notice to decouple from external LLM providers during initial launch.

B. Spaced Repetition Mistake Notebook (`/dashboard/ai-coach/mistakes`)

- **File Location:** `src/app/dashboard/ai-coach/mistakes/page.tsx`
- **Access Guard:** `AI_COACH` Tier
- **UI Components:**
  - Daily Mistake Card Queue: Displays questions that the student answered incorrectly in previous CBT exams, scheduled for revision today using the SuperMemo-2 (SM-2) algorithm.
  - Interactive Answer Selector (`A`, `B`, `C`, `D`)
  - "Submit Answer" button
  - Post-submission Feedback Panel: Immediate correctness status, explanation, and new scheduled review interval (e.g. *"Reviewed! Next review in 6 days"*).
  - "Next Question" button
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Select Answer (A/B/C/D)`	Selects answer option	`setSelectedOption(opt)`
`Submit Answer Button`	Submits review and updates SM-2 interval	`submitReview(mistakeId, option)`

Button / Trigger	Action	Target / Handler
`Next Question Button`	Advances to next queued mistake	`handleNext()`

- **APIs Invoked:**
  i. `GET /api/mistakes/today?user_id={userId}`
    - **Trigger:** On page mount
    - **Response:** Array of pending mistake objects where `next_review <= NOW()` .
  ii. `POST /api/mistakes/review`
    - **Trigger:** On `Submit Answer` click
    - **Request Body:** `{ "mistake_id": "<mistakeId>", "selected_answer": "B" }`
    - **Response:**

```
{
  "status": "reviewed",
  "is_correct": true,
  "interval": 6,
  "next_review": "2026-09-15T00:00:00Z"
}
```

C. Exam Readiness & College Predictor (`/dashboard/ai-coach/readiness` & `/college-predictor`)

- **File Location:** `src/app/dashboard/ai-coach/readiness/page.tsx` , `college-predictor/page.tsx` , `ReadinessGauge.tsx` , `CollegePredictorCard.tsx`
- **Access Guard:** `AI_COACH` Tier
- **UI Components:**
  - Circular SVG Exam Readiness Gauge (0–100%)
  - Readiness Breakdown: Accuracy Score, Time Management Score, Consistency Score, Topic Coverage
  - AI College Predictor Card: Predicted Rank, Eligible IIT/NIT/AIIMS colleges based on historical cutoffs (Safe, Probable, Reach)
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Recalculate Score Button`	Refetches latest readiness calculation	`fetchReadiness()`
`Explore College Cutoffs`	Expands category details	`setCategoryFilter(...)`

- **APIs Invoked:**
  i. `GET /api/ai-coach/readiness/{userId}`
    - **Trigger:** On mount
    - **Response:**

```
{
  "readiness_score": 78,
  "predicted_rank": 3420,
  "confidence_interval": "± 400",
  "subject_breakdown": {
    "Physics": 82,
    "Chemistry": 85,
    "Mathematics": 67
  }
}
```

D. Adaptive Daily Study Plan (`/dashboard/ai-coach/study-plan`)

- **File Location:** `src/app/dashboard/ai-coach/study-plan/page.tsx` & `src/components/AI/StudyPlanCard.tsx`
- **Access Guard:** `AI_COACH` Tier
- **UI Components:**
  - Daily Study Plan Task List: Revision Target, Practice Target, Test Recommendation

- Task checkboxes to mark tasks as completed

- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Task Checkbox`	Marks daily task complete	Local task state
`Start Recommended Task`	Deep-links to relevant test or DPS	`/dashboard/dpp`

- **APIs Invoked:**
  i. `GET /api/ai-coach/study-plan/today?user_id={userId}`
    - **Trigger:** On mount
    - **Response:** Daily plan JSON generated by Celery Beat at 5:00 AM IST.

3.6 Admin Management Console (`/admin/*`)

> **Access Restriction:** Protected by `proxy.ts` (Requires `auth_token=admin` cookie). Non-admin users are immediately redirected to `/login` .

A. Command Center Overview (`/admin`)

- **File Location:** `src/app/admin/page.tsx`
- **Access Guard:** Admin Only
- **UI Components:**
  - KPI Stat Cards: Total Revenue, Active Students, AI Agent Accuracy %, Total Questions
  - System Health & Accuracy Area Chart (`recharts`)
  - Quick Action Buttons: "Refresh Data", "Create Test", "Manage Students"
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Refresh Data Button`	Triggers refetch of dashboard metrics	`fetchDashboardData()`
`Create Test Quick Link`	Navigates to Test Management	`/admin/tests`

B. Student Directory & Profile Control (`/admin/students` & `[studentId]`)

- **File Location:** `src/app/admin/students/page.tsx` and `src/app/admin/students/[studentId]/page.tsx`
- **Access Guard:** Admin Only
- **UI Components:**
  - Students Directory Table: Name, Email, Target Exam, Class, Tier Badge (`Free` / `Pro` / `AI Coach`), Streak, Actions
  - Search & Filter Bar: Filter by Exam (`JEE Main` / `NEET`) or Tier
  - Detail Page (`[studentId]`):
    - Full Profile Card
    - Tier Override Selector (`Free -> Pro -> Al_Coach`)
    - "Update Tier" button
    - "Reset Streak" button
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`View Student Row`	Navigates to individual student control	`router.push('/admin/students/' + student.id)`
`Update Tier Button`	Changes student subscription tier	`handleTierUpdate(newTier)`
`Reset Streak Button`	Resets student streak counter to 0	`handleResetStreak()`

- **APIs Invoked:**
  i. `GET /api/admin/students` (Lists all student accounts)
  ii. `GET /api/admin/students/{studentId}` (Fetches single student record)
  iii. `PATCH /api/admin/students/{studentId}`

- **Trigger:** On tier change or profile update
- **Request Body:** `{ "tier": "AI_Coach" }`
- **Response:** Updated student record.

C. Test & DPS Management Center (`/admin/tests`)

- **File Location:** `src/app/admin/tests/page.tsx`
- **Access Guard:** Admin Only
- **UI Components:**
  - Active Tests Table: Title, Exam Pattern, Total Time, Scheduled Date, Active Switch toggle, Delete button
  - Manual Test Creation Form / JSON Drag-and-Drop Uploader:
    - Test Title `<input />`
    - Exam Pattern selector (`JEE_MAIN` , `JEE_ADVANCED` , `NEET`)
    - Test Type selector (`Mock` , `DPS`)
    - Time Limit `<input />` (minutes)
    - JSON Question File Uploader (Validates question schema: `question_text` , `options` , `correct_answer` , `subject` , `topic`)
    - "Create Test" button
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Active Toggle Switch`	Toggles test visibility to students	`handleToggle(testId)`
`Delete Test Button`	Permanently deletes test and its questions	`handleDelete(testId)`
`JSON File Drop Zone`	Parses and validates questions JSON	`processFile(file)`
`Create Test Button`	Dispatches manual test creation	`handleCreateTest()`

- **APIs Invoked:**
  i. `GET /api/admin/tests`
    - **Trigger:** On mount
    - **Response:** Array of all existing tests.
  ii. `PUT /api/admin/tests/{testId}/toggle`
    - **Trigger:** On switch toggle
    - **Response:** `{ "status": "toggled", "is_active": true|false }`
  iii. `DELETE /api/admin/tests/{testId}`
    - **Trigger:** On delete confirm
    - **Response:** `{ "status": "deleted" }`
  iv. `POST /api/admin/tests/create_manual`
    - **Trigger:** On `Create Test` submit
    - **Request Body:**

```
{
  "title": "JEE Main Physics Final Mock",
  "type": "Mock",
  "exam_pattern": "JEE_MAIN",
  "time_limit": 180,
  "admin_id": "<adminId>",
  "questions_json": [ ... ]
}
```

    - **Response:** `{ "status": "created", "test_id": "<uuid>", "questions_count": 75 }`

D. Coupon Management (`/admin/coupons`)

- **File Location:** `src/app/admin/coupons/page.tsx`
- **Access Guard:** Admin Only

- **UI Components:**
  - Active Coupons Table: Code, Discount %, Max Discount Amount, Status, Delete button
  - New Coupon Creation Card: Code input, Discount percentage input, Max discount input, "Create Coupon" button
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Create Coupon Button`	Submits new coupon	`handleCreate()`
`Delete Coupon Button`	Deletes discount coupon	`handleDelete(couponId)`

- **APIs Invoked:**
  i. `GET /api/admin/coupons` (Fetches active coupons list)
  ii. `POST /api/admin/coupons`
    - **Request Body:** `{ "code": "PRAGATI50", "discount_percent": 50, "max_discount_amount": 2000 }`
  iii. `DELETE /api/admin/coupons/{id}` (Deletes coupon)

E. Global Platform Settings (`/admin/settings`)

- **File Location:** `src/app/admin/settings/page.tsx`
- **Access Guard:** Admin Only
- **UI Components:**
  - Maintenance Mode Switch
  - Free Test Quota `<input />` (Number of free mock tests allowed before paywall)
  - Allow Registrations Switch
  - "Save Settings" button
- **Buttons & Interactive Elements:**

Button / Trigger	Action	Target / Handler
`Save Settings Button`	Persists global configuration	`handleSaveSettings()`

- **APIs Invoked:**
  i. `GET /api/admin/settings` (Fetches current settings)
  ii. `POST /api/admin/settings`
    - **Request Body:** `{ "maintenance_mode": false, "free_test_limit": 2, "allow_registrations": true }`

4. Master Button-to-API Cross Reference Matrix

Button / UI Element	Page Location	Trigger Type	HTTP Method & Endpoint	Payload / Parameters
Sign In	`/login`	Form Submit	`POST /api/auth/login`	`{ email, password }`
Register & Login	`/signup`	Form Submit	`POST /api/auth/register`	`{ name, email, password }`
Reset Password	`/forgot-password`	Form Submit	`POST /api/auth/forgot-password`	`{ email }`
Proceed to Payment	`/checkout`	User Click	`POST /api/billing/create-order`	Query: `?user_id=...&plan_tier=...`
Razorpay Verification	`/checkout`	SDK Callback	`POST /api/billing/verify-payment`	`{ razorpay_order_id, razorpay_payment_id, ..`
Start CBT Exam	`/dashboard/cbt`	Navigation	`GET /api/tests/{examId}`	Query: `?user_id=...`

Button / UI Element	Page Location	Trigger Type	HTTP Method & Endpoint	Payload / Parameters
Fetch Questions	`/cbt/[examId]`	Page Mount	`GET /api/tests/{examId}/questions`	Path: `{examId}`
Final Submit Exam	`/cbt/[examId]`	Modal Confirm	`POST /api/attempts/submit`	`{ test_id, user_id, answers, ... }`
Submit Mistake Review	`/dashboard/ai-coach/mistakes`	User Click	`POST /api/mistakes/review`	`{ mistake_id, selected_answer }`
Send AI Doubt	`/dashboard/ai-coach/tutor`	Form Submit	`POST /api/ai-coach/chat/stream`	`{ user_id, message }`
Upload Avatar	`/dashboard/settings`	File Input	`POST /api/users/{id}/avatar`	Form-Data: `file: <image>`
Save Profile	`/dashboard/settings`	User Click	`PUT /api/users/{id}`	`{ name, target_exam, school, state }`
Toggle Test Status	`/admin/tests`	Switch Toggle	`PUT /api/admin/tests/{id}/toggle`	Path: `{id}`
Delete Test	`/admin/tests`	Button Click	`DELETE /api/admin/tests/{id}`	Path: `{id}`
Create Manual Test	`/admin/tests`	Form Submit	`POST /api/admin/tests/create_manual`	`{ title, exam_pattern, questions_json, ... }`
Update Student Tier	`/admin/students/[id]`	Dropdown Change	`PATCH /api/admin/students/{id}`	`{ tier: "Al_Coach" }`
Create Coupon	`/admin/coupons`	Form Submit	`POST /api/admin/coupons`	`{ code, discount_percent, ... }`
Delete Coupon	`/admin/coupons`	Button Click	`DELETE /api/admin/coupons/{id}`	Path: `{id}`
Save Admin Settings	`/admin/settings`	User Click	`POST /api/admin/settings`	`{ maintenance_mode, free_test_limit, ... }`

5. Codebase Cleanup Audit & Phase 2 Roadmap

5.1 Cleaned & Removed Legacy Files

In accordance with production standards, all dead/legacy files, log dumps, and temporary development artifacts have been hard-deleted:

- `backend/fix.py`, `backend/fix_from_source.py`, `backend/fix_ions.py`: One-time scripts for manual question text extraction and repair.
- `backend/seed_dpps.py`, `backend/seed_dpps_mock.py`, `backend/seed_dpps_real.py`, `backend/seed_from_dpps.py`: Legacy seed scripts superseded by `seed.py`.
- `backend/pragativeer-prd.pdf`, `backend/test.pdf`, `backend/test2.pdf`: Temporary test binaries.
- `backend/pragativeer.db`: Abandoned SQLite file (PragatiVeer exclusively uses PostgreSQL).
- `backend/pdfs_to_parse/`: 15.3 MB of raw PDFs removed (already extracted into `admin_uploads/`).
- `backend/pdf_parser.py`: Unused stub file removed to prevent dead code clutter.
- `backend.log`, `backend_logs.txt`, `full_logs.txt`: Obsolete runtime logs.

- `alter.sql` , `userlist.txt` , `CLAUDE.md` : Temporary scratch and SQL debug files.
- `create_policies.py` , `fix_icons.py` , `generate_jee_json.py` : One-time generation/migration scripts.

5.2 Preserved Academic & Testing JSON Assets

As required for manual verification, test runs, and format specifications, the following question datasets remain actively preserved in the codebase:

- `jee_advanced_full_90q.json` : Comprehensive 90-question JEE Advanced sample dataset for exam testing.
- `neet_180_questions.json` : Comprehensive 180-question NEET UG sample dataset for exam testing.
- `admin_json_samples.md` : Official JSON schema specifications and samples for Admin question uploader.
- `neet_sample_questions.md` : NEET sample question reference and validation format.
- `admin_uploads/` : 4 Daily Practice Sheets (Biology, Chemistry, Mathematics, Physics Chapter 4 JSONs) seeded directly into the CBT engine.

5.3 Decoupled Dependencies (`requirements.txt`)

The following heavy external packages were removed from `backend/requirements.txt` :

- `openai==1.14.0` : Decoupled to eliminate reliance on external LLM keys during initial rollout.
- `PyMuPDF==1.23.26` : Decoupled to prevent native C-library compile overhead in production.

5.4 Production-Safe AI Modules

- `backend/llm.py` : Cleanly stubbed with an async generator returning a friendly coming-soon notice.
- `backend/worker.py` : Retains exclusively the active 5:00 AM IST Celery Beat daily study plan scheduler.
- `POST /api/admin/tests/upload` : Saves PDF to MinIO and guides administrators to use `/api/admin/tests/create_manual` until Phase 2 is provisioned.

Document compiled and verified for PragatiVeer v2 — Production Release