🧭 Mendel Academy – Developer Understanding Document

Project Type: Educational Web App (Medical Competitive Exam Preparation Platform)
Target Launch Date: 31 October 2025
Developed By: Shopno Ecommerce Pvt. Ltd.
Developers:

- 👩‍💻 Node.js Developer – Backend (APIs, Database, Logic)
- 💻 Next.js Developer – Frontend (UI, Pages, Integration)

🌍 1. Project Overview

Mendel Academy is an online platform for **medical students preparing for competitive exams** like USMLE Step 1, 2, 3 and more.
Students can **buy Question Banks, join Live Lectures**, and **purchase Pre-recorded Lecture Packages**.

The system allows:

- Secure online payments
- Subscription validity (3, 5, or 6 months)
- Auto-expiry of access after the validity period
- Blog publishing for updates and tips
- Centralized admin control panel

👥 2. User Roles

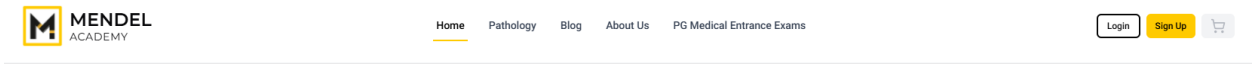
User Type	Description

👨‍🎓 **Student**	Registers, logs in, purchases subscriptions, views lectures, reads blogs
👨‍💼 **Super Admin**	Manages all content (courses, question banks, blogs, payments)
👩‍🏫 **Instructor (Future Scope)**	Adds or hosts courses (to be added later)

🔄 3. Full User Flow (A–Z)

Step 1: Home Page

- User opens `mendelacademy.com`
- Sees main sections:



- Highlights include:
    - Featured Question Banks
    - Live Courses by Dr. Kishore Managoli
    - Latest Blogs
    - Testimonials and About section

Step 2: Registration / Login

- User clicks **Login / Register**
- Fills form (Name, Email, Password)
- After successful login:
    - Redirected to **Dashboard**

    - Shows active courses, validity, and next steps

Step 3: Question Bank Flow

- From the top menu → **Question Banks**
- Each Question Bank card shows:
    - Title, Description, Price, Duration (e.g., 3 Months)
    - “Enroll Now” button
- User clicks **Enroll Now** → Payment popup (Razorpay or Stripe)
- After successful payment:
    - System stores transaction
    - Auto-generates expiry date
    - Sends confirmation email
- Dashboard shows:
    - Purchased Question Banks with expiry countdown

Step 4: Pre-recorded Lectures Flow

- User goes to **Pre-recorded Lectures**
- Sees available lectures with:
    - Duration (3, 5, 6 months)
    - Price
    - “Buy Now” button

- After purchase:
    - Video access unlocked in **My Lectures**
    - Access restricted by expiry date
- Video player integrated (via Vimeo)
    - Only logged-in, subscribed users can view
    - Access auto-blocked after expiry

Step 5: Combo Offers Flow

- Combos show multiple lectures bundled with discounts
- Each combo card displays:
    - Included courses
    - Original & discounted price
- On purchase:
    - All lectures within combo are unlocked
    - Validity same for all in the combo

Step 6: Live Course Flow

- User goes to **Live Courses**
- Sees upcoming sessions by Dr. Kishore Managoli
- Clicks **Subscribe Now**
- After payment:

    - Request sent to Admin
    - Admin approves → system auto-sends Zoom link via email
- On the session day:
    - User joins directly through the provided link

Step 7: Blogs Flow

- User clicks **Blogs** in top menu
- Sees list of blog articles:
    - Title, Image, Short Description
- Click “Read More” → full blog opens
- Blogs are managed from the Admin Panel:
    - Add / Update / Delete

Step 8: Admin Panel Flow

Admin Login: `/admin`

Admin Dashboard Includes:

1. **Dashboard Overview**
    - Total Users, Active Subscriptions, Total Sales
2. **Question Banks**
    - Add, Edit, Delete Question Banks
3. **Pre-recorded Courses**

- Upload via Vimeo (store video ID)
- Set validity (3/5/6 months) and pricing

4. **Live Courses**
    - Schedule sessions
    - Approve student subscriptions
    - Send Zoom link via email
5. **Combos**
    - Create bundles of multiple lectures with discounts
6. **Blogs**
    - Add, Update, Delete blog posts
7. **Payments**
    - View and search transactions
8. **Settings**
    - Manage API keys (Zoom, Vimeo, Email templates)

⚙️ 4. Technical Overview (Simplified)

Component	Technology	Description
Frontend	Next.js	User Interface, Pages, Routing
Backend	Node.js (Express or Nest.js)	APIs, Authentication, Payment Logic
Database	MySQL or MongoDB	Store users, subscriptions, courses, blogs
Video Hosting	Vimeo	Secure paid video streaming

Payment Gateway	Razorpay / Stripe	Online payments
Email System	Nodemailer (Gmail SMTP)	For confirmations and Zoom links
Deployment	Vercel (Frontend), Render/Railway (Backend)	Hosting

---

🧩 5. Module Overview (Developer Responsibility Split)

Module	Node.js Developer	Next.js Developer
Setup	Server, DB config	Layout, Navbar, Routing
Authentication	Login, JWT, API	Login, Signup Pages
Question Banks	CRUD + Payment	Listing, Enroll Flow
Pre-recorded	Validity, Vimeo logic	Player + Dashboard
Live Course	Approvals, Zoom link mail	Subscription UI
Payment	Gateway setup, Webhooks	Payment popup UI
Blogs	CRUD APIs, Image upload	Blog list & single post
Admin Panel	All APIs + Auth	Dashboard pages

🧠 6. Developer Workflow

Daily Working Pattern

1. Pick 1 module from Jira / task list.
2. Backend dev builds the API & logic.
3. Frontend dev integrates it into the UI.
4. Test end-to-end (data, logic, design).

5. Mark task as complete in tracking tool.
6. End-of-day: update on progress / blockers.

🗓️ 7. Weekly Goal Plan

Week	Goals	Expected Outcome
Week 1 (Oct 14–20)	Setup + Auth + Question Banks	Working base project + Question Bank module
Week 2 (Oct 21–25)	Payments + Pre-recorded + Admin Panel	Payment success + Admin CRUD ready
Week 3 (Oct 26–31)	Live Courses + Blogs + Final Testing	Full working site + ready for client demo

---

✅ 8. Final Pre-Launch Checklist

Area	Check
User Registration / Login	✅ Working
Payment Flow	✅ Success + Webhook verified
Subscription Expiry	✅ Auto restricts access
Vimeo Integration	✅ Secure + Authenticated
Zoom Live Flow	✅ Email link sent after approval
Admin CRUD	✅ All sections working
Blogs	✅ List + Single + SEO setup
Responsiveness	✅ Mobile + Desktop tested
Deployment	✅ Vercel + Render working
Backup / DB Export	✅ Ready for handover

🏁 Final Notes

- Keep code modular and clean.
- Store Vimeo video IDs, not full links.
- Use environment variables for API keys.
- Always check expiry logic before production.
- Send test payment and Zoom mail before launch.

.