PROMPT — BUILD “FleaxovA” FULL-STACK PLATFORM

Role

You are a Senior Full-Stack Software Architect and Lead Engineer with strong expertise in modern web platforms, secure payment systems, and scalable SaaS architectures.
You think in terms of clean architecture, modular services, and production-ready code.

Project Overview

Build FleAxovA, a payment-only freelancing platform exclusively for students.

FleAxovA enables students to:

- **Create verified freelancer profiles**
- **List paid services only (no free work)**
- **Receive payments securely through the platform**
- **Deliver work digitally**
- **Build credibility through reviews and completed orders**

The platform must be fully full-stack, covering frontend, backend, database, authentication, payments, and deployment-ready structure.

Design & UI/UX Requirements

- **Theme: Corporate, premium, minimal**
- **Color Palette:**
  - **Primary: White**
  - **Secondary: Black**
  - **Accent: Subtle gray tones only**
- **Style: Clean, professional, enterprise-grade (no flashy colors)**
- **Typography: Modern sans-serif, readable, business-oriented**
- **UX Principles:**
  - **Simple navigation**
  - **Student-friendly workflows**
  - **Clear pricing and payment visibility**
  - **Responsive (desktop + mobile)**

Technical Stack (MANDATORY)

Frontend

- **Framework: React.js (Vite setup)**
- **Styling: CSS / Tailwind (clean corporate usage)**
- **State Management: React Context or Redux (where appropriate)**
- **Routing: React Router**
- **API Communication: Axios / Fetch**
- **Authentication Handling: JWT-based**

Backend

- **Runtime: Node.js**
- **Framework: Express.js**
- **Architecture:**
  - **MVC / Service-based architecture**
  - **Clean separation of routes, controllers, services, and models**
- **Authentication:**
  - **JWT (Access + Refresh tokens)**
  - **Role-based access (Student, Admin)**
- **Security:**
  - **Password hashing**
  - **Input validation**
  - **Secure headers**

Database

- **Primary DB: MongoDB**
- **ODM: Mongoose**
- **Data Models:**
  - **User**
  - **Student Profile**
  - **Services**
  - **Orders**
  - **Payments**
  - **Reviews**
  - **Withdrawals**

Core Functional Modules (FULL-STACK)

1. Authentication System

- **Student signup & login**
- **Email/phone verification**
- **Secure JWT sessions**
- **Password reset**

2. Student Freelancer Profiles

- **Skill listing**
- **Service pricing (mandatory paid services)**
- **Portfolio section**
- **Ratings & reviews**
- **Profile verification badge**

3. Service Marketplace

- **Service creation (paid only)**
- **Category-based discovery**

- **Search & filter**
- **Service detail pages**

4. Order & Workflow System

- **Order placement**
- **Payment confirmation before work starts**
- **Order status tracking**
- **Delivery submission**
- **Client approval**
- **Automatic completion**

5. Payment System (CRITICAL)

- **Payment-only platform (no cash, no free work)**
- **Integration with Indian payment gateway (UPI/cards)**
- **Platform commission logic**
- **Wallet balance for students**
- **Secure withdrawal requests**
- **Admin approval for withdrawals**

6. Review & Trust System

- **Post-order reviews**
- **Star ratings**
- **Anti-spam controls**
- **Visibility rules**

7. Admin Panel

- **User management**
- **Service moderation**
- **Payment monitoring**
- **Withdrawal approvals**
- **Platform analytics**

Business Rules (STRICT)

- **Only students can register as freelancers**
- **All services must be paid**
- **Payment is mandatory before order confirmation**
- **Platform deducts commission automatically**
- **Withdrawals allowed only after order completion**
- **No direct contact sharing before payment**

Code & Development Standards

- **Write clean, commented, production-ready code**
- **Follow industry naming conventions**

- **Modular, reusable components**
- **Environment variable usage**
- **Error handling with meaningful responses**
- **API documentation where needed**

Output Expectations from AI

- **Proper folder structure (frontend + backend)**
- **Step-by-step code generation**
- **Each module built method-by-method**
- **Clear explanations alongside code**
- **No placeholder logic — real implementations**

Final Goal

Deliver a fully functional, scalable, professional full-stack freelancing platform that feels corporate, trustworthy, and student-focused — ready for real-world deployment.