

# EduConnect Product Report

## Purpose and Vision

- Mission: Connect students with qualified teachers for online learning through a seamless marketplace with professional profiles, reliable scheduling, and built-in video sessions.
- Vision: A trusted platform where teachers showcase experience like LinkedIn, students discover the right match, and classes run smoothly with integrated video and bookings.

## Target Users and Personas

- Students/Learners: Need to find teachers by subject/skill, book sessions, and join classes easily.
- Teachers/Instructors: Need to showcase credentials, list gigs, manage bookings, and run classes.
- Admins: Oversee users, activities, and platform health.

## Core Value Propositions

- Professional Profiles: LinkedIn-inspired Experience/Education/Work sections to build trust.
- Frictionless Booking: Clear request → accept/reject flow with real-time status.
- Integrated Classes: Automatic Jitsi links for secure, private video sessions.
- Modern UX: Responsive design, searchable gigs, intuitive dashboards.

## Scope (MVP)

- Auth & Onboarding: Registration/login with role selection and isOnboarded gating.
- Profile System: Student/Teacher profiles with editable Experience, Education, Work.
- Gig Discovery: Search/browse teacher gigs with filters.
- Bookings: Pending/Accepted/Rejected flow; dashboards for both roles.
- Video Calls: Auto-generated Jitsi rooms on acceptance; join from dashboards.
- Admin Console: User list and activity monitoring.

- Uploads: Cloudinary images/videos; gig thumbnails.

## Key Features

- Authentication & Onboarding
  - Role-based access (student, teacher, admin)
  - isOnboarded and marketingSource captured; gated redirects to /onboarding
  -
- Profiles (LinkedIn-style)
  - Experience, Education, Work sections; badges for skills/subjects/languages
  - Public teacher profile page /teacher/[id]/profile
  - Editable sections (add/edit/delete) for owners
  -
- Gig Management & Discovery
  - Teachers create gigs (title, description, subjects, hourly rate, thumbnail)
  - Students browse/search; gig detail with “View Full Profile”
  -
- Booking System
  - Students request booking on a gig
  - Teachers manage in dashboard: Requests (Pending), Accepted, Rejected
  - Status updates with optimistic UI; dashboard stats
  -
- Video Calls (Jitsi)
  - On acceptance, backend generates roomId and meetingLink (<https://meet.jit.si/<roomId>>)
  - Both parties see “Join Class” on relevant bookings
  - Dedicated meeting page with Jitsi embed and access validation
  -
- Admin Management
  - User list and activity feed with filters (action, user, target)
  - Activity logging from critical flows (auth, bookings, gigs, etc.)
  -
- Uploads

- Cloudinary-backed image/video uploads
- Gig thumbnail upload updates gig document and previews

## How It Works (User Flows)

- Onboarding Flow
  - New user logs in → if `isOnboarded=false` or missing name/role → `/onboarding`
  - Completes name, role, marketing source → dashboard
- Student: Discover & Book
  - Browse gigs → view gig → “View Full Profile” if needed
  - Click “Book” → submit request → status: pending
  - On acceptance → “Join Class” visible → join via meeting page (Jitsi)
  - Post-session: leave review (if enabled)
- Teacher: Manage Requests & Teach
  - Dashboard shows Requests with serial numbering (#1, #2, ...)
  - Accept/Reject → if accepted, system sets `meetingLink` and `meetingRoomId`
  - “Join Class” from Accepted tab (highlighted for today’s classes)
- Admin: Monitor & Moderate
  - Access users table; filter and review activity logs
  - Investigate anomalies via activity detail
- 

## Data and APIs (Overview)

- Models
  - User: `name`, `email`, `role`, `isOnboarded`, `marketingSource`, `profile sections`
  - Gig: `teacherId`, `title`, `subjects`, `rate`, `thumbnail`
  - Booking: `studentId`, `teacherId`, `gigId`, `date/time`, `price`, `status`  
(`pending/accepted/rejected`), `meetingLink`, `meetingRoomId`

- Activity: actor, action, targetType, targetId, metadata, timestamp
- Representative Endpoints
  - Auth: register/login
  - Users: PUT /api/users/me (name, role, isOnboarded, marketingSource)
  - Gigs: CRUD and search
  - Bookings: GET /api/bookings?status=..., PUT /api/bookings/:id (status updates), GET /api/bookings/dashboard/stats
  - Uploads: POST /api/uploads/image, POST /api/uploads/video, POST /api/uploads/gig-thumbnail?gigId=X
  - Admin: GET /api/admin/users, GET /api/admin/activities
- 

## Non-Functional Requirements

- Security: JWT, role guards, access validation for meetings
- Performance: Responsive UI, efficient queries, pagination
- Reliability: Clear error states; idempotent updates; robust upload handling
- Accessibility: Keyboard navigation, contrast, ARIA where needed
- Observability: Activity logs, structured errors

## Success Metrics

- Time-to-first-booking
- Booking acceptance rate
- Session completion rate (join events)
- Profile completeness for teachers
- Admin-flagged issues per week

## Risks & Assumptions

- Jitsi dependency and network constraints
- Teacher supply vs student demand balance
- Payment processing scope and compliance (phase-specific)

## Roadmap

- MVP: Profiles, gigs, bookings, Jitsi, onboarding, admin monitoring, uploads
- Phase 2: Payments & wallet UX, calendar sync, group classes, messaging, advanced reviews

## Acceptance Criteria (MVP)

- Students can find a teacher, request a booking, and join a Jitsi class after acceptance.
- Teachers can accept/reject, see meeting links, and join classes.
- Public teacher profile shows LinkedIn-style sections.
- Admin can list users and review activity logs.
- Uploads support gig thumbnails and demo videos.