**Larnik E-Learning Platform**

Functionality Document

Powered by: Meadow Agriculture Pvt. Ltd.

Tech Stack: MERN Stack (MongoDB, Express.js, React.js, Node.js)

Hosting: Hostinger VPS / Cloud

Version: Initial Launch

Larnik E-Learning Platform  
Complete Orientation & Functionality Document

Powered by: Meadow Agriculture Pvt. Ltd.

Tech Stack: MERN Stack (MongoDB, Express.js, React.js, Node.js)

Hosting: Hostinger VPS/Cloud

Version: Initial Launch (Live Classes Skipped)

🔰 Project Lifecycle - Phase-wise Execution Flow

Phase 1: UI/UX Design – All dashboards, mobile-friendly layout, Figma designs  
Phase 2: Frontend Development – React.js frontend, routing, responsive interface  
Phase 3: Backend Development – Node.js APIs, MongoDB schemas, Razorpay integration  
Phase 4: Local Testing – QA team performs module-wise testing  
Phase 5: VPS Deployment – Hostinger deployment, SSL, domain setup, Nginx  
Phase 6: Final QA Testing – End-to-end testing with bugs fixed  
Phase 7: Approval Phase – Internal approval by Governance + Admin  
Phase 8: Credentials Submission – All role-based logins shared  
Phase 9: Final Architecture Report – Diagrams, workflows, SOP documentation

🧩 Project Dashboards – Detailed Functional Overview

1. Super Admin Dashboard

Purpose: Central control system to manage the entire platform, users, content, finance, and sub-admins.  
  
Features:  
- Manage users: Teachers, Students, Partners, Career Cell, Writers, Reviewers, Universities  
- Course approvals with feedback loop via Review Board  
- Design certificates and upload signatories  
- Bulk notifications via email/SMS  
- Finance: monitor and approve settlements  
- Export analytics and reports

2. University Dashboard & Sub-Roles

Purpose: Universities can upload/manage courses, assign staff, control certification and monitor student activity.  
  
Roles:  
- Super Admin: Assign university staff roles  
- Course Manager: Upload/edit courses  
- Finance Officer: View revenue, request settlements  
- Documentation Manager: Upload MoUs, tie-up papers  
- Certification Officer: Issue approval  
- Performance Analyst: Monitor student pass/fail  
- Referral Manager: Track course promotion  
  
Flow:  
- Create Course → Add Videos (1–10+) → Notes → MCQs  
- Certificate issued only on 33%+ score  
- Custom certificate with university branding

3. Teacher Dashboard

Purpose: Teachers manage content, earnings, student engagement.  
  
Features:  
- Profile setup with ID, Bank, UPI, photo  
- Course Upload Flow: Category → Sub-Category → Course  
- Add videos, notes, MCQs, assignments  
- Earnings and feedback monitoring  
- Request settlement via wallet panel

4. Student (Learner) Dashboard

Purpose: Students learn, give exams, and earn certificates.  
  
Features:  
- Register/login via OTP  
- Purchase course, watch videos in sequence  
- Attempt exam (pass = 33%+)  
- Download certificate with QR/ID  
- Submit course reviews  
- Referral system enabled optionally

5. Referral Partner Dashboard

Purpose: Promote and earn commission on course sales.  
  
Features:  
- Register or convert from student  
- Share course referral links  
- Earn % commission (1% to 10%)  
- Wallet tracking and request settlement

6. Sub-Admin Dashboards

Blog Manager: Approves and schedules articles  
  
Finance Manager: Approves settlements, logs revenue  
  
Career Cell: Adds jobs, guides, webinars  
  
Governance Body: Validates MoUs, certificates  
  
Review Board: Reviews course content with remarks

UI/UX Guidelines

- Use Figma for wireframing, mobile-first layout  
- Drag-drop uploads, modal preview for videos/certificates  
- Graph-based reports for earnings and students

Developer Guidelines

- React.js + Tailwind for frontend  
- Node.js + MongoDB backend with JWT authentication  
- Razorpay and secure file uploads  
- Certificate generation with PDFKit

Final Checklist

✅ Dashboards Functional  
  
✅ UI/UX Ready  
  
✅ Frontend & Backend Integrated  
  
✅ OTP, Payment, Upload Tested  
  
✅ VPS + Domain + SSL Live  
  
✅ Logins Issued  
  
✅ Architecture Report Submitted

🔰 Project Lifecycle

1. UI/UX Design → Wireframes for all dashboards using Figma with responsive design.  
2. Frontend Development → React.js-based mobile-first implementation.  
3. Backend Development → Node.js RESTful APIs, MongoDB schemas.  
4. Local Testing → Internal QA and debugging.  
5. VPS Deployment → Setup on Hostinger, SSL + domain linked.  
6. Final Testing → Full testing of all roles & modules.  
7. Approval Phase → Admin and Governance body approval.  
8. Login Handover → All login credentials distributed.  
9. Architecture Submission → Documentation and diagrams handed over.

📚 Course Category & Upload Process (Like Coursera/Great Learning)

All course uploads must follow a hierarchical and structured format:  
  
1. CATEGORY (Main Subject Area):  
 Examples:  
 - Agriculture  
 - Computer Science  
 - Business & Entrepreneurship  
 - Health & Wellness  
 - Personal Development  
 - Finance & Accounting  
 - Languages (English, Hindi, etc.)  
 - Environment & Sustainability  
  
2. SUB-CATEGORY:  
 Examples (under Computer Science):  
 - Web Development  
 - Artificial Intelligence  
 - Data Science  
 - Cybersecurity  
  
3. COURSE CREATION:  
 Title, Description, Duration, Target Audience, Prerequisites, Tags  
  
4. MODULES:  
 Each course will be broken into modules (min 1 to max 20+)  
  
5. VIDEO UPLOAD STRUCTURE:  
 - Format: .MP4, 720p minimum resolution  
 - Each module will contain 1–5 videos (avg 5–20 minutes each)  
 - Drag-drop UI to upload files  
 - Video order is maintained and must be watched sequentially  
 - Instructor must upload Thumbnail & Title per video  
  
6. SUPPORTING MATERIALS:  
 - Notes in PDF (1 per module minimum)  
 - Optional Assignments  
 - MCQ Bank: 5–15 questions per course for certification  
  
7. CERTIFICATE RULE:  
 - Student must score ≥ 33% in MCQ to receive certificate  
 - Certificate is auto-generated with QR code and unique ID

1. UI/UX Design – Wireframes and responsive views for all dashboards.  
2. Frontend Development – React.js-based component UI.  
3. Backend Development – Node.js APIs with MongoDB.  
4. Local Testing – QA testing of modules.  
5. VPS Deployment – Hosted on Hostinger with SSL.  
6. Final QA Testing – Complete end-to-end functional test.  
7. Approval Phase – Governance & admin-level final checks.  
8. Credentials Submission – Role-wise login credentials handed over.  
9. Architecture Report – Complete documentation with flow diagrams.

📚 Course Structure and Upload (Coursera Style)

Hierarchy:  
  
1. Category (e.g., Agriculture, Tech, Finance, Health)  
2. Sub-category (e.g., AI, Data Science, Organic Farming)  
3. Course Title + Description + Thumbnail + Tags  
4. Modules – Each course must have at least 1 to 20+ modules  
5. Video Upload – Each module may include 1–5 videos (MP4, HD)  
6. Notes – PDF files per module (mandatory)  
7. Assignments – Optional per module  
8. MCQs – 5–15 randomized questions for certification  
9. Exam – Pass mark set at 33%  
10. Certificate – Auto-generated if passed (with QR + ID)

🧩 Dashboard Roles & Features

1. Super Admin Dashboard

- User management: Teachers, Students, Partners, Writers, University  
- Course control: Approve/reject courses via review board  
- Finance: Approve settlement for teachers/referrals/universities  
- Certificate design upload: Signatory & format manager  
- Reports: Revenue, enrollments, analytics (exportable)

2. University Dashboard & Roles

- University can upload courses (like teacher)  
- Revenue split: % same as teacher (customizable)  
- Staff dashboard roles:  
 a) Super Admin: Assign staff  
  
 b) Course Manager: Upload/edit content  
  
 c) Finance Officer: Settlement handling  
  
 d) Certificate Officer: Approval of pass students  
  
 e) Document Manager: MoU uploads  
  
 f) Analyst: Monitor student pass/fail stats  
- Certificate issuance after 33% exam pass

3. Teacher Dashboard

- Upload course: Category → Sub-category → Course  
  
- Add video (MP4), notes (PDF), MCQs, assignment  
  
- Dashboard shows enrollments, ratings, earnings  
  
- Settlement wallet for income withdrawal  
  
- Feedback & review viewing

4. Student Dashboard

- Register/login via OTP  
  
- Browse & purchase course  
  
- Watch videos sequentially  
  
- Attempt MCQs (auto check)  
  
- Pass: ≥ 33% = Certificate with QR + ID  
  
- Certificate vault & referral system included

5. Referral Partner Dashboard

- Register or convert from student  
  
- Share referral links, earn % commission:  
  
 - 1–10 users: 1%  
  
 - 11–20: 2.5%  
  
 - 21–40: 5%  
  
 - 41–50+: 10%  
  
- Wallet system + settlement request

6. Sub-Admin Dashboards

- Blog Manager: Review & publish blogs  
  
- Finance Manager: Handle payouts and logs  
  
- Career Cell: Add job blogs, internships  
  
- Governance Body: MoU, certification validation  
  
- Review Board: Evaluate course content quality

🎨 UI/UX Guidelines

- Figma-based design system  
  
- Mobile-first responsive UI  
  
- Upload wizards with progress bar  
  
- Sequential video viewer with lock/unlock  
  
- Certificate preview/download modal

💻 Developer Guidelines

- Frontend: React.js + Tailwind  
  
- Backend: Node.js with Express  
  
- JWT auth + role-based access  
  
- Razorpay integration  
  
- Secure media uploads & PDFKit for certificate

📋 Final Checklist

✅ UI/UX Finalized  
  
✅ All Dashboards Functional  
  
✅ Backend APIs Working  
  
✅ OTP + Payment Tested  
  
✅ Hosting & SSL Done  
  
✅ Logins Handed Over  
  
✅ Final Architecture Report Submitted