MedLearn Project Overview

A learning management system (LMS) tailored for independent pharmacies. Provides features for course delivery, user and company management, business inquiries with approval workflows, and an admin dashboard for monitoring.

Technology Stack - Runtime: Node.js (ES modules) - Web framework: Express - Database: MySQL via mysql2/promise - Session management: express-session, cookie-parser - Validation: express-validator with custom DNS MX and NPI checks, Google reCAPTCHA filecite turn15file0 L5-L13 - Email: Mailgun API (mailgun.js, form-data) - Frontend tooling: Parcel bundler, Tailwind CSS - Additional libraries: intl-tel-input, bcrypt, cors filecite turn15file0 L15-L19

Key Environment Variables - MYSQL_HOST, MYSQL_USER, MYSQL_PASSWORD, MYSQL_DATABASE - SERVER_PORT (default 3000) - SESSION_SECRET - MAILGUN_API_KEY, MAILGUN_DOMAIN, EMAIL_FROM, MAILGUN_API_BASE_URL - RECAPTCHA_SECRET - FRONTEND_URL (for onboarding link generation)

Directory Structure MedLearn/ ├── db/ # Database utilities & SQL │ ├── database.js # MySQL connection pool │ ├── queries.sql # Common SQL queries │ └── tables.sql # Table definitions and triggers ├── src/ │ ├── server/ │ │ ├── server.js # Express app setup and middleware │ │ ├── routes/ # API endpoint definitions │ │ ├── daos/ # Data-access objects for each table │ │ └── utils/ # Helpers (mailer, validators, MX checker) │ └── client/ │ ├── public/ # Unauthenticated static pages and assets │ │ ├── index.html, courses.html, inquiries.html, signin.html │ └── secure/ # Auth-protected UI │ ├── dashboard.html # Admin dashboard with alerts and widgets │ └── welcome.html # Onboarding for first-time users ├── package.json # Scripts and dependencies ├── nodemon.json # Dev server watcher config ├── tailwind.config.js # Tailwind CSS setup └── .gitignore

filecite turn15file0 L28-L36

Backend Highlights - server.js loads environment variables, configures JSON parsing, cookies, and 8-hour sessions (secure flag off by default) - requireAuth and requireAdmin middleware protect secure routes - Routes: • GET /api/session – returns current session user or 401 • POST /api/users/login – authenticate and start session • POST /api/users/logout – destroy session • POST /secure/api/users – create user (with optional admin role) • GET /secure/api/users – list users (admin only) • DELETE /secure/api/users/:id – delete user • POST /api/companies – create company • GET /api/companies – list companies • DELETE /api/companies/:companyName – remove company and associated users • POST /api/inquiries – submit business inquiry (includes Mailgun confirmation) • GET /api/inquiries/unapproved – list pending inquiries for admin • PATCH /api/inquiries/:id/approve – approve inquiry, create invite, update status • PATCH /api/inquiries/:id – general status updates (denied, pending) • GET /api/npi-validation/:npi – external NPPES registry validation

Data Access Objects - companiesDao.js – CRUD for companies - usersDao.js – user creation (bcrypt password hashing) and login checks - inquiriesDao.js – store inquiries, list unapproved, update status and inviteID - invitesDao.js – create invites (tokenHash, expirationTime), lookup by tokenHash, mark used

Utilities - mailer.js – Mailgun helpers for sending inquiry and invite emails - validationSchema.js – express-validator schemas for all routes - mxChecker.js – DNS MX record lookup for email domain validation

Frontend Highlights - Public pages use Parcel for bundling and Tailwind CSS for styling - inquiries.html features a form with built-in NPI, phone, and reCAPTCHA checks - signin.html handles user login, dashboard.html and welcome.html are behind auth - Admin UI (admin.html & admin.js) dynamically renders pending inquiries, supports row selection for details, and Approve/Deny actions with live DOM updates

Database Schema - companies (companyID, companyName) - users (id, firstLogin, companyID, firstName, lastName, role, email, password_hash) - inquiries (inquiryID, email, firstName, lastName, npi, inquiryType, phoneNumber, numOfUsers, msg, approved, inviteID, createdAt DATETIME DEFAULT CURRENT_TIMESTAMP, deniedAt) - invites (inviteID, tokenHash, tokenUsed BOOLEAN, expirationTime DATETIME DEFAULT (CURRENT_TIMESTAMP + INTERVAL 72 HOUR))

Build & Run Scripts - npm run clean – remove .parcel-cache and dist - npm run dev – clean, start Parcel (watch) and server (nodemon) in parallel - npm run watch – Parcel watch only - npm run server – start Express server only - npm run build – production asset bundling - npm start – production server (NODE_ENV=production)

Notable TODOs - Integrate dynamic course listings from database API - Flesh out dashboard widgets with real metrics - Migrate onboarding flow into React frontend (long-term) - Improve email templates with Tailwind-compatible inlining - Add pagination and search to admin inquiries table