Auth Backend (MERN) — Project Summary

About the Project

A production-ready authentication backend for MERN apps, deployed on Render. It supports email-based OTP registration, secure login with JWT, and password resets, with strong validation, rate limiting, and bot protection (Cloudflare Turnstile + honeypot).

Live: https://auth-backend-3eq4.onrender.com
Tech: Node.js, Express 5, MongoDB Atlas (Mongoose)

Core Features

- **Registration via Email OTP**
    - Start register → send 6-digit OTP (10-minute expiry, 60-second resend throttle)
    - Verify OTP
    - Complete register → set hashed password (bcrypt) and return JWT
- **Login**
    - Email + password → JWT (HS256, default 7-day expiry)
- **Forgot/Reset Password**
    - Send reset OTP → verify OTP → set new password
- **Operational**
    - Health check endpoint for uptime and deployment verification

---

External Services Used

- **Cloudflare Turnstile** (siteverify) — human verification on critical forms (login/register/reset)
- **Gmail SMTP via Nodemailer** — OTP email delivery (App Password)
- **MongoDB Atlas** — managed database

Key Dependencies

- **Express 5.1** — HTTP server and routing (with proper 404 + error middleware ordering)
- **Mongoose** — MongoDB models and querying
- **Zod** — strong input validation (email and password complexity)
- **bcrypt** — secure password hashing
- **jsonwebtoken** — stateless auth tokens (JWT)
- **express-rate-limit** — per-route rate limits to curb abuse
- **cors** — strict origin allow-list for frontend integration
- **nodemailer** — transactional OTP emails

Security & Anti-Abuse

- **Cloudflare Turnstile** on sensitive endpoints (server-side verification)
- **Rate limiting** per route (tighter on OTP and login)
- **OTP protections**: 10-minute expiry, 60-second resend throttle
- **Password storage**: bcrypt hashing
- **JWT**: HS256, short default expiry (7d), strong random secrets
- **CORS**: explicit allow-list driven by environment variables
- **HTTP hardening**: Helmet headers
- **Honeypot** : hidden field + optional time-trap to silently drop basic bots
- **Consistent JSON error handler** and Express-5-safe 404s

Architecture (Overview)

- REST API (Express 5) → MongoDB Atlas (Mongoose)
- Email pipeline via Gmail SMTP (Nodemailer)
- Human verification via Cloudflare Turnstile (frontend site key, server secret)
- Deployed on Render (Node 20 LTS), health check path /health, auto-deploy from GitHub

---

Frontend Integration

- Frontend calls the backend using a base URL env var
- Turnstile widget returns a token; client includes it with form submissions
- Backend verifies Turnstile token before processing sensitive actions
- CORS is preconfigured to allow the deployed frontend and local development

---

Purpose: Power secure auth for my MERN apps with strong validation, anti-abuse, and a smooth deployment, suitable for real-world use.