WORKFLOW AUTOMATION PLATFORM (OverlayMVP)

WHAT IT IS

- AI-powered Chrome extension + web dashboard to record and guide interactive web workflows (e.g., onboarding and SOP walkthroughs).

WHO IT'S FOR

- Department managers at SMBs (10-500 employees) who need to capture "how we do this" processes and help teammates follow them consistently.

KEY FEATURES

- Chrome extension records user actions and captures screenshots
- Guided walkthrough overlay on live websites
- AI-generated step labels/instructions (Celery background task)
- Auto-healing support when UI elements change
- Dashboard to sign up/login and review workflows + step details
- Company/team features: invites, roles, and permission checks
- Notifications + health endpoints for workflow status/visibility

HOW IT WORKS (ARCHITECTURE + DATA FLOW)

Components (repo evidence):

- Extension (Manifest V3): popup UI, background service worker, content scripts
- Backend API: FastAPI routers for auth, workflows, steps, screenshots, healing, company, invites, notifications, health, and users
- Database: SQLAlchemy with SQLite by default (DATABASE_URL=sqlite:///./app.db)
- Async jobs: Celery worker with Redis broker (AI labeling)
- Storage: screenshots in S3 if configured; otherwise local backend/screenshots
- Dashboard: React + Vite web app (proxies /api to backend in dev)

Flow:

1) Record in extension -> upload steps/screenshots -> backend stores records
2) Extension triggers workflow processing -> Celery labels steps via AI service
3) Dashboard reads workflows/steps via REST API; extension runs walkthrough UI

HOW TO RUN (MINIMAL GETTING STARTED)

```
1) Install JS deps:        npm install
2) Backend venv + deps:    cd backend && python -m venv venv && \
                           source venv/bin/activate && pip install -r requirements.txt
3) Configure env:          cp backend/.env.example backend/.env  (set JWT_SECRET_KEY,
                           DATABASE_URL, REDIS_URL, ANTHROPIC_API_KEY as needed)
4) DB migrations:          cd backend && alembic upgrade head
5) Start services (3 tabs):
   - Backend API:          cd backend && source venv/bin/activate && \
                           uvicorn app.main:app --reload --port 8000
   - Dashboard:            cd dashboard && npm run dev   (http://localhost:3000)
   - Celery worker:        cd backend && source venv/bin/activate && \
                           celery -A app.celery_app worker --loglevel=info
6) Load extension:         cd extension && npm run build; then load extension/dist/
                           in Chrome at chrome://extensions (Developer mode).
```