

```

# Mahaseel (■■■■■■■) - 45-Day Technical Daily Plan
**Scope:** Build an MVP mobile marketplace connecting Sudanese farmers and buyers.
**Stack:** Flutter (mobile), FastAPI (backend), PostgreSQL (DB), SQLAlchemy + Alembic
(ORM/migrations), JWT (auth), Google Maps (location), WhatsApp deep link (chat), Docker & Docker
Compose, Railway/Render (hosting).
**Repo Layout (monorepo):**
...

mahaseel/
  backend/
  mobile/
  infra/
  docs/
...

**Conventions:** GitHub flow, Conventional Commits, .env per service, pre-commit hooks, CI checks
(lint, test, build).

---

## Day 1 - Project Bootstrap
- Create monorepo folder structure (`mahaseel/{backend,mobile,infra,docs}`).
- Initialize git repo (`git init`, default branch `main`).
- Add root `.editorconfig`, `.gitignore`, `README.md` (vision, scope, stack).
- Decide package managers & versions; document in `docs/tech-decisions.md`.
- Create issue templates & labels on GitHub (bugs, feature, chore).

## Day 2 - Backend Skeleton (FastAPI)
- Create `backend/app/main.py` with `/healthz` route.
- Add `pyproject.toml` or `requirements.txt` (fastapi, uvicorn, pydantic, sqlalchemy, alembic, python-
multipart, passlib[bcrypt], python-jose, pillow).
- Configure logging (JSON formatter).
- Add config loader (`pydantic-settings`) and `.env.example`.
- Write `Makefile` targets: `run`, `format`, `lint`, `test`.

## Day 3 - Database & Migrations
- Start local Postgres (Docker) and connect from backend.
- Define models: `User`, `Crop`, `Order`, `Media`.
- Initialize Alembic and generate first migration.
- Add DB seed script `scripts/seed.py` (admin user).

## Day 4 - Auth (Phone-first, no SMS yet)
- Implement phone-based signup/login endpoints.
- `POST /auth/register {phone, name}`.
- `POST /auth/login {phone}` → returns mock OTP (dev), JWT after verify.
- Add Bearer auth middleware.
- Unit tests for auth flows.

## Day 5 - Crop CRUD (Core)
- Endpoints: `POST /crops`, `GET /crops`, `GET /crops/{id}`.
- Fields: name, type, qty, price, unit, location {lat,lng}, notes.
- Ownership checks & Pydantic validation.
- Tests + Postman collection start (`docs/postman.json`).

## Day 6 - Media Uploads
- Add image upload: `POST /media` multipart → store in `backend/uploads/` (dev).

```

- Serve via static mount `/static/*`.
- Link `Media` to `Crop` (main image).
- Validate size/dimensions; downscale with Pillow.

Day 7 – Location

- Location helper and reverse geocode optional.
- Store `state`, `locality`, `address` with `lat/lng`.
- Filter by `state` in `/crops`.

Day 8 – WhatsApp Deep Link

- Endpoint: `GET /contact/{seller_id}/whatsapp` → `https://wa.me/<phone>?text=<encoded>`.
- Document UX flow in `docs/flows.md`.
- E2E test: create seller → create crop → generate deep link.

Day 9 – Backend Hardening

- CORS config for mobile app.
- Error handlers (422/404/500) with consistent JSON shape.
- Rate limiting (slowapi).
- Simple roles: `seller`, `buyer`, `admin`.

Day 10 – Dockerization (Backend)

- Create `Dockerfile` (multi-stage) and `docker-compose.yml` (app + db).
- Healthchecks for app & db.
- Makefile: `make up`, `make down`, `make logs`.

Day 11 – Flutter App Bootstrap

- `flutter create mobile` with package IDs & app name (Mahaseel).
- Foldering: `lib/{features,core,services,widgets}`.
- Install packages: `dio`, `riverpod`, `bloc`, `freezed`, `json_serializable`, `go_router`, `intl`, `geolocator`, `google_maps_flutter`, `image_picker`, `shared_preferences`, `flutter_secure_storage`.
- Base theme & RTL localization (ar default).

Day 12 – Auth Screens

- Screens: Welcome, Login (phone), Mock OTP (dev), Session persistence.
- API service for auth (Dio client with interceptors).
- Store JWT securely (flutter_secure_storage).

Day 13 – Crop List UI

- Home/List screen with cards (image, name, price, location).
- Infinite scroll/pagination; pull-to-refresh.
- Empty/error states and skeleton loaders.

Day 14 – Add Crop UI

- Form: images (picker + camera), name, type, qty, price, unit, location picker.
- Submit to `POST /crops`; handle validation errors.
- Local draft save (offline).

Day 15 – Crop Details & Contact

- Detail screen: gallery, seller info, map preview.
- WhatsApp CTA button using deep link.
- Share card (Android share intent).

Day 16 – Maps Integration

- Google Maps widget on details; tap to open native Maps.

- Request location permission; handle denial gracefully.
- Cache tiles when possible.

Day 17 – State Management & Offline

- Introduce app-wide state (Bloc/Riverpod).
- Offline cache for list/details using `hive` or `sqflite`.
- Retry queue for failed creates when online.

Day 18 – QA Round 1 (Mobile + Backend)

- Test on low-end Android device; profile jank.
- Fix top UX issues (fonts, tap targets, RTL).
- Stabilize API contracts; update Postman.

Day 19 – CI/CD Basics

- GitHub Actions:
 - Backend: lint (ruff), test (pytest), build image, push to GHCR.
 - Mobile: format, analyze, build debug APK as artifact.
- Add status badges in README.

Day 20 – Logging & Metrics

- Backend: request/response logs (PII-safe).
- Prometheus metrics (optional).
- Mobile: Firebase Crashlytics, basic analytics events.

Day 21 – Security & Config

- Audit dependencies (pip-audit, `flutter pub outdated`).
- Secrets policy: `.env` locally, `.env.example` in repo only.
- Backup/restore for Postgres; daily `pg_dump` script.

Day 22 – Staging Environment

- Deploy backend to Railway/Render with managed Postgres.
- Seed staging data (5 sellers, 10 crops, 3 buyers).
- Point mobile `BASE_URL` to staging; build internal APK.

Day 23 – Internal Testing

- Distribute APK to 10 pilot users.
- Feedback form; issues to GitHub.
- Hotfix critical crashes.

Day 24 – Performance Pass

- Backend: add indexes; optimize N+1 queries.
- Mobile: image compression; list virtualization tuning.
- Manual UX pass for fast tap paths.

Day 25 – Features Polish

- Filters: crop type, state, price range.
- Sort: newest, price asc/desc.
- Persist last filters in local storage.

Day 26 – Ratings (MVP)

- Add `Rating` model (1-5 stars).
- Endpoint + UI to submit/view average rating per seller.
- Anti-abuse: one rating per buyer per seller/crop.

Day 27 – Orders (Lightweight)

- Non-binding "Intent to Buy": `POST /orders` (crop_id, qty, note).
- Seller statuses: `new|chatting|agreed|closed`.
- Local notifications for status changes.

Day 28 – Docs & Support

- Write `docs/user-guide.md` (screenshots), `docs/api.md` (OpenAPI link).
- In-app support page (FAQ + email/WhatsApp link).

Day 29 – Beta Prep

- Play Console setup (listing, privacy policy, content rating).
- Generate signed release key; secure handling (never commit).
- Build `flutter build appbundle` for Play testing.

Day 30 – Beta Launch

- Publish to Closed Testing; invite testers.
- Monitor Crashlytics and backend logs.
- Landing page with download link & WhatsApp contact.

Day 31 – Post-Beta Fixes

- Address top 10 tester issues.
- Improve empty/error states (Arabic-first).
- Add "Report Listing" (abuse flag).

Day 32 – Data & Insights

- Simple admin dashboard to view users/crops/orders.
- Export CSV endpoint for crops/orders.

Day 33 – Mobile Money (Design Only)

- Draft API contracts for MTN/Zain/Bankak (mock providers).
- Payment method selector UI (disabled).

Day 34 – Logistics Hook (Design Only)

- Define `Carrier` model & webhook placeholders.
- "Request Transport Quote" → email/WhatsApp to partner.

Day 35 – Accessibility & Locale

- Full RTL support and accessible contrast.
- Add en/ar localization files; default to `ar`.

Day 36 – Growth Utilities

- Referral deep links (`?ref=PHONE`).
- Handle app deep links to open specific crop.

Day 37 – Hardening & Backups

- Automated DB backups on staging (daily), retention policy.
- Disaster recovery runbook in `docs/runbooks/dr.md`.

Day 38 – E2E Tests

- Playwright/Flutter integration: login, create crop, list, details, WhatsApp tap.
- Backend smoke tests in CI with ephemeral DB.

Day 39 – Release Candidate

- Cut `v0.9.0-rc` tag; feature freeze.

- Regression checklist; fix blockers.

Day 40 – Public Beta

- Open Testing on Play.
- Announce on FB/WhatsApp farmer groups.
- Monitor onboarding funnel metrics.

Day 41 – Observability

- Add Sentry to backend (optional).
- Correlation IDs, dashboards for 4xx/5xx, latency, DB CPU.

Day 42 – Scalability Prep

- CDN for static images (Cloudflare/S3).
- Background jobs queue for thumbnails (RQ/Celery).

Day 43 – Security Review

- JWT expiry/refresh, authZ checks.
- Validate uploads (MIME), size limits.
- Pen-test checklist; fix medium/high findings.

Day 44 – Final Polish

- App icons, splash, onboarding tips (2-3 screens).
- Store screenshots & short promo video.
- Review Arabic copy (clear Sudan context).

Day 45 – Soft Launch (Sudan)

- Tag `v1.0.0` and publish.
- Small ad budget; collect conversion metrics.
- Retro & plan next 45 days (payments/logistics).

Deliverables Summary

- Mobile app (Flutter) + Backend (FastAPI) + Postgres.
- Dockerized services + CI/CD + staging environment.
- Docs: API, user guide, runbooks, tech decisions.
- Play testing tracks, analytics, crash reporting, backups.