**📅 Month 1 Plan (Setup Phase)**

**Week 1 – Project Foundation**

✅ Goals: Create repo, choose stack, set up base environment.

* Create **GitHub repo** with branches (main, dev, feature/\*).
* Decide CI/CD tool (GitHub Actions is good for now).
* Setup **frontend skeleton**:
  + Create React Native app (npx react-native init OneHealthGuardian).
  + Add navigation (react-navigation).
  + Add UI kit (Tailwind RN or NativeBase).
* Setup **backend skeleton**:
  + Create Node.js + Express project (npm init).
  + Add routes (/auth, /health, /journal).
  + Setup FastAPI service folder for ML tasks.
* Setup **database**:
  + Dockerize PostgreSQL + MongoDB locally.
  + Create initial schema (users, journal\_entries).

📌 Deliverable: Repo with frontend, backend, and DB containers working locally.

**Week 2 – Backend & Auth**

✅ Goals: Basic backend APIs and authentication.

* Implement **user authentication (JWT)** in Node.js.
* Add **Postgres models**:
  + Users (id, name, email, password, language).
* Add **Mongo models**:
  + Journal entries (id, userId, entry, sentimentScore).
* Create routes:
  + POST /auth/signup → register user.
  + POST /auth/login → login and get JWT.
  + GET /user/me → profile info.
* Connect **backend to DB** (Sequelize/Prisma for Postgres, Mongoose for Mongo).

📌 Deliverable: Backend running with working signup/login + DB persistence.

**Week 3 – Frontend + API Connection**

✅ Goals: Connect frontend with backend.

* Setup **auth screens** in React Native:
  + Signup / Login UI.
  + Save JWT securely (AsyncStorage).
* Connect **API calls** from frontend → backend.
* Create simple **dashboard screen** after login.
* Add **basic state management** (Redux Toolkit or Context API).
* Setup **.env config** for API base URLs.

📌 Deliverable: User can signup/login in the app → see profile dashboard.

**Week 4 – AI Service Skeleton**

✅ Goals: Add ML service foundation (no real AI yet).

* Setup **FastAPI microservice** (separate container).
* Add dummy endpoints:
  + POST /symptom-checker → returns { diagnosis: "placeholder" }.
  + POST /sentiment → returns { mood: "neutral" }.
* Connect Node.js backend → FastAPI service (proxy requests).
* Connect frontend button → call dummy AI endpoint (shows result).
* Add **logging (Winston/ELK)** for backend & ML service.

📌 Deliverable: Frontend button → backend → ML service → dummy AI response shown in app.

**✅ End of Month 1:**

You’ll have:

* Repo + CI/CD pipeline.
* Frontend skeleton (React Native).
* Backend skeleton (Node.js + JWT auth).
* Database schemas (Postgres + Mongo).
* FastAPI ML service with dummy endpoints.
* Working signup/login → dashboard → AI dummy response.