📋 MediHelp+ Team Assignment Document

---

1. Backend (Django) — 4 Members

Member	Task	Deadline	Notes
Bitsuan Abate	Build core AI API integration (OpenAI, Gemini) for Symptom Checker `/api/symptom-checker` and Chat `/api/chat-health-assistant`	Day 1–3	AI interaction logic and error handling
Wondifraw Terefe	Build `/api/daily-coaching`, `/api/local-health-alerts`, and `/api/nearby-hospitals` endpoints	Day 2–4	Health education, alerts, geolocation integrations
Tewodros Anteneh	Set up backend monitoring (Uptime Kuma, Healthchecks.io, PM2), build `/api/healthz`, and implement AI API monitoring middleware (latency and failure detection)	Day 1–5	Ensuring uptime monitoring, backend health resilience, and AI reliability checks
Mesud Melaku Asfaw	Create `/api/first-aid-guidance/{case}` endpoint (pulls static files or DB text) Design a fault-tolerant Django API with dynamic case-matching (regex validation, 404/503 error handling, and static/DB fallback logic). Optimize response caching (15-min TTL) and integrate with Uptime Kuma/Sentry for SRE compliance. Delegate content population to team members (WHO-standard JSON/DB entries) and frontend integration (React error states).	Day 2–3	First aid guidance delivery

---

2. Frontend (React) — 6 Members

Member	Task	Deadline	Notes
Ruth Akalu	Build Symptom Checker Form (connect to `/api/symptom-checker`)	Day 1–3	Form validation and result rendering
Eyerusalem Gashaw Misganaw	Build Health AI Chatbot Interface (connect to `/api/chat-health-assistant`)	Day 2–4	Simple chat interface with real-time responses
Bayisa Daba Gutu	Build Daily Health Coaching Card and Local Alerts Banner	Day 2–4	Dashboard components for user coaching and alerts
Bezawit Taye	Build First Aid Mode Selection + Voice Guidance (SpeechSynthesis API integration)	Day 3–5	First aid user flow
Dagim Demissie	Build Nearby Hospitals Map/Link Feature (connect to `/api/nearby-hospitals`)	Day 3–5	Maps or basic links to health centers
Mesud Ahmed	Integrate Sentry for error logging, Web Vitals tracking, and setup frontend crash recovery modals	Day 1–3	Frontend monitoring and error resilience

---

3. Additional Group Tasks (Shared)

Task	Members	Deadline	Notes
Chaos Testing Day (simulate backend + AI failures)	Backend + Frontend teams together	Day 6	Practice downtime recovery
Discord Alert Configuration (Healthchecks.io + Kuma)	Backend team	Day 2	Uptime and incident alerting setup

Final Deployment Setup (PM2 scripts, production mode settings)	Backend team	Day 5–6	Prepare final deployment stability

---

4. Project Milestone Timeline

Day	Backend Focus	Frontend Focus
Day 1	Setup Kuma, PM2, Healthz endpoint	Setup Sentry, Healthchecks.io for frontend
Day 2–3	Build APIs: Symptom Checker, First Aid, Coaching, Alerts	Build Symptom Form, Chatbot interface
Day 4	Integrate AI APIs (GPT, Gemini) into backend	Connect frontend UIs to backend APIs
Day 5	Setup database pings, automated server recovery	Final UI feature polish
Day 6	Conduct Chaos Testing (simulate outages and fixes)	Handle frontend crash recovery testing
Day 7	Final fixes, presentation polishing, submit project	Final fixes, presentation polishing

---

5. Final Notes

- All backend APIs should be lightweight and fast.
- Frontend should gracefully handle API failures without crashing.
- Alerts through Discord must be instant and clear.
- Focus first on Symptom Checker, First Aid Mode, and Health Coaching — these are core MVP features.
- Prioritize keeping the user experience smooth even if AI services (OpenAI, Gemini) slow down or fail.

✅ Ready for Assignment!

Each member now has a clear task, timeline, and deliverables. Stay synced daily, use small commits, and keep communication active on Discord for quick issue resolution.