1. Backend (Node.js + MongoDB)

- Set up MongoDB connection (use Mongoose for schema modeling)
- Create models:
  - User → name, email, password hash, role (patient/family/doctor)
  - Medication → userId, name, dosage, schedule, status (taken/missed)
  - ReminderLog → medicationId, timestamp, status
  - Doctor → name, email, password hash, specialization, patients (array of User references)
  - Report → patientId, doctorId, title, description, fileUrl, createdAt, updatedAt
- Set up REST API endpoints:
  - POST /register, POST /login
  - POST /medications, GET /medications, PUT /medications/:id, DELETE /medications/:id
  - GET /dashboard/:userId
  - **Doctor Report endpoints:**
    - POST /reports → upload report (Doctor)
    - GET /reports/:patientId → list patient reports (Doctor + Patient)
    - GET /reports/:id → view report (Doctor + Patient)
    - PUT /reports/:id → edit report (Doctor)
    - DELETE /reports/:id → delete report (Doctor)

---

2. Frontend (Next.js)

- Pages & components:
  - /login, /signup → authentication forms
  - /medications → list, add, edit, delete meds
  - /dashboard → history + charts
  - **Doctor Reports:**
    - /doctor/reports → Doctor view: add/upload, edit, delete reports
    - /patient/reports → Patient view: list & download reports

- Use **Axios** or **Fetch** to call backend API

---

3. Reminders (Node.js Cron Jobs)

- Cron job checks every minute/hour for upcoming medication
- Send email (SendGrid) for reminders/alerts
- If not marked “taken” → send alert to family member

Jira Board: Smart Medicine Reminder with Family Alerts (Web)

Epic

Title: Smart Medicine Reminder with Family Alerts (Web)

Sprint 1 — Week 1: Research & Setup

Task	Assignee
Research Next.js basics	Person A
Research Node.js + MongoDB integration	Person B
Research SendGrid API (Email)	Person C
Research Chart.js / Recharts	Person D
Create GitHub repo + base Next.js project	Person A
Create Figma wireframe mockups	Person E
Document research findings	All

Sprint 2 — Week 2: Authentication & Profiles

Task	Assignee
Setup MongoDB User model	Person A
Create API routes: POST /register & POST /login	Person A
Implement signup/login forms	Person E
Setup MongoDB Medication model	Person B
Create CRUD API skeleton for medications	Person B
Setup notification system skeleton (SendGrid)	Person C
Prepare dashboard API structure	Person D
Setup Doctor & Report models	Person C
Create API skeleton for Doctor Reports	Person C

Sprint 3 — Week 3: Medication Management

Task	Assignee
Integrate auth with medication DB access	Person A
Complete CRUD endpoints & validation	Person B
Integrate cron job skeleton to call backend for meds	Person C
Start fetching data from backend for dashboard	Person D
Build medication UI (Add/Edit/Delete)	Person E
Doctor: Upload & view reports UI	Person C
Patient: View/download reports UI	Person D

Sprint 4 — Week 4: Reminders & Notifications

Task	Assignee
Integrate auth + family/doctor linking	Person A
Ensure medication schedule logic is correct	Person B
Cron jobs: send reminders & update MongoDB	Person C
Display basic dashboard tables	Person D
Connect medication UI with reminders & taken/missed flags	Person E
Enable Doctor Report notifications via email	Person C

Sprint 5 — Week 5: Dashboard Backend & Logic

Task	Assignee
API integration for dashboard filters	Person A
Ensure medication data integrity	Person B
Test notification triggers & edge cases	Person C
Compute compliance % & generate dashboard API endpoints	Person D
Build UI components for charts & tables	Person E
Include Doctor Report data in dashboard	Person D

Sprint 6 — Week 6: Dashboard Frontend & UI

Task	Assignee
Help integrate backend APIs with frontend	Person A
Assist in connecting CRUD forms with frontend	Person B
Assist in testing reminders & notifications	Person C
Implement chart visualization + filters	Person D
Finalize responsive layout + navigation	Person E
Add Doctor Reports tabs/views in dashboard	Person C & D

Sprint 7 — Week 7: Integration & Testing

Task	Assignee
Merge frontend + backend	All
Test authentication, CRUD, notifications, dashboard	All
Test Doctor Report upload/download & email notifications	All
Fix bugs	All

Sprint 8 — Week 8: Deployment & Documentation

Task	Assignee
Write backend & API documentation	Person A
Test database integrity + final checks	Person B
Ensure reminders, notifications, and Doctor Reports are reliable	Person C

Task	Assignee
Validate dashboard data correctness	Person D
Deploy app to Vercel + UI final adjustments	Person E