Project Concept Application:

A note taking application for a primary care physician when meeting with patients to record notes concerning symptoms and referencing a database of recommended drugs to alleviate those symptoms.

Assumptions- single developer (worst case)

- Stack: Flask API, SQLite, simple HTML/JS (or lightweight React) front-end.
- Scope: note taking + symptom chips → generic drug suggestions; no dosing, no eRx, no billing.
- Definition of Done includes basic unit tests on backend endpoints and manual UI check.

Stakeholders

Sprint 0 — Project & DB bootstrap (5-7 days)

1. As a developer, I want a runnable project skeleton so I can add features safely.
   AC: flask run serves health endpoint; env config; venv instructions.
   Estimate: 2-3 days
2. As a developer, I want the schema and seed data created on first run so I can test quickly.
   AC: schema. SQL auto-applied; seed symptoms & meds present.
   Estimate: 2–3 days

Sprint 1 — Patients & Encounters (5-7 days)

3. As a clinician, I can create/search/select a patient to start a note.
   AC: API: POST/GET /patients; basic UI search by name/MRN; validation.
   Estimate: 2-3 days
4. As a clinician, I can start an encounter and capture SOAP fields to record the visit.
   AC: API: POST /encounters with patient id, CC, S/O/A/P; persists.
   Estimate: 2-3 days
5. As a clinician, I can add allergies to a patient, so unsafe meds are flagged.
   AC: API: POST/GET /patient allergies; UI add/remove list.
   Estimate: 1-2 days

Sprint 2 — Symptoms & Recommendations (3-7 days)

6. As a clinician, I can pick symptoms via chips/autocomplete to structure the complaint.
   AC: API: GET /symptoms; UI multiselect; stored on encounter.
   Estimate: 1-2 days
7. As a clinician, I see generic drug suggestions when symptoms are selected to speed planning.
   AC: API: GET /recommendations? Symptoms=...&allergies=...; sorted by priority; max 6.
   Estimate: 1-2 days
8. As a clinician, I can add a suggested med to the Plan text with one click to reduce typing.
   AC: “Add to Plan” appends “generic — class (OTC/Rx). Rationale.”
   Estimate: 1-2 days
9. As a clinician, I see clear guardrails on suggestions, so I don’t over-rely on the tool.
   AC: Banner: “Informational only. Clinician decides.” Static caution text displayed.
   Estimate: 1 day

Sprint 3 — Safety, UX polish (3-7 days)

10. As a clinician, I see allergy conflicts highlighted and filtered to prevent unsafe picks.
    AC: Any med containing allergy keyword is hidden; warning badge shown if all filtered.
    Estimate: 1-2 days
11. As a clinician, I can quickly edit Plan text including free notes after adding suggestions.
    AC: Rich text area; newline bullets; keyboard shortcuts (CTRL Enter to save).
    Estimate: 2-3 days
12. As a clinician, I can save a draft and resume to avoid data loss mid-visit.
    AC: Auto-save every 10s or on blur; draft recovered on reload.
    Estimate: 2-3 days

Sprint 4 — History & Output (3-7 days)

13. As a clinician, I can see past encounters for a patient to maintain continuity.
    AC: GET /encounters? Patient _id=...; list shows date, CC; click to view.
    Estimate: 2-3 days
14. As a clinician, I can copy/export a clean visit note for my EMR.
    AC: “Copy to clipboard” and “Download PDF” (simple template); timestamped.
    Estimate: 2-3 days
15. As a clinician, I can print a one-page summary for the paper chart when needed.
    AC: Print-CSS stylesheet; hides UI chrome.
    Estimate: 1-2 days

Sprint 5 — Admin & Data Curation (3-7 days)

16. As an admin, I can manage the symptom list to match clinic language.
    AC: CRUD for symptoms (behind a basic admin password).
    Estimate: 2-3 days
17. As an admin, I can manage the medications table (generic/class/OTC/Rx).
    AC: CRUD for medications; validation for duplicate generics.
    Estimate: 2-3 days
18. As an admin, I can edit symptom med recommendation mappings to refine suggestions.
    AC: CRUD for symptom _med _recs with priority & rationale.
    Estimate: 1-2 days

Optional (Backlog / Nice-to-haves)

19. Contraindication toggles (pregnancy, CKD, GI bleed) narrow suggestions.
    AC: Simple checkboxes apply static filters.
    Estimate: 1-2 days
20. User accounts & audit log (who changed what).
    AC: Basic login; Audit Log table; reads/writes tracked.
    Estimate: 1-2 days

21. Tag favorites (per user) for commonly used plans.
    AC: “ ⭐ Save as snippet”; insert via picker.
    Estimate: 1-2 days

Summary timeline (single dev, “likely” case)

- Sprint 0: 5-7 days
- Sprint 1: 5-7 days
- Sprint 2: 3-7 days
- Sprint 3: 3-7 days
- Sprint 4: 3-7days
- Sprint 5: 3-7days

Total MVP (Sprints 0–4): ~19–28 working days
With Admin tools (Sprint 5): ~22–42 working days