

QUESTION: \* Helps during \*\*handover/handoff\*\* between nurses.

okay so would that delete the pre-existing chart in the software, once they click export data?

Great clarification.

No—clicking “Export Data” should not delete any charted information. It simply creates a copy (PDF/TXT) of what's already stored in the system.

#### 💡 Why It Shouldn't Delete:

Continuity of Care: Nurses often need to refer back to previous notes from earlier shifts.

Audits or Disputes: Healthcare notes may be reviewed days or weeks later.

Legal/Compliance: Charted information may need to be stored for years, depending on policy—even if you're not integrated with a hospital system yet.

Multiple Handoffs: The next nurse may export the same data again, or just view it in-app.

#### 💡 What You Should Do Instead:

Action	Result
Nurse charts via voice	Data saved to the database
Nurse clicks **"Export PDF"**	App generates a *copy* of that data in PDF
Original charted data	**Remains stored in the system**
Nurse returns tomorrow	Can still see all their previous notes
Another nurse logs in	Can see all past notes by room/bed

#### ⌚ Optional Features You Could Add Later:

Mark as "Reviewed" or "Handed Over" (but still keep it stored)

"Archive older notes" after a set period, but don't delete

Soft-delete by admin if data needs to be cleared after export in special cases (e.g. training simulations)

Final Recommendation

Do not delete data after export.

Let export be a non-destructive action — just a snapshot for communication or recordkeeping.

Let me know if you'd like a UX wireframe showing how this export-review-handoff flow could look visually.

QUESTION: "Archive older notes"\*\* after a set period, but don't delete

yeah, cause I was thinking, what if a patient is discharged, what happens to the information, where would it go

Great question — and you're absolutely thinking like someone who wants to build for scale and real-world adoption. Here's how you can handle patient discharge and data lifecycle without deleting anything prematurely, while keeping things organized and compliant.

 Goal:

When a patient is discharged, their charted notes should remain safely stored, but moved out of the active workflow (so nurses don't see clutter from inactive patients).

What Happens When a Patient is Discharged

 Suggested Flow:

Patient is marked as "Discharged"

Could be triggered:

Manually (by the nurse or admin)

Automatically (via external data integration, in the future)

After X hours/days of inactivity

Notes for that room/bed get moved to "Archived Records"

Not deleted — just categorized differently

Still queryable (for reviews, audits, summaries, or disputes)

That room/bed is "reset" for new patients

New notes begin under a new patient ID

Past data is no longer shown in the "active view" for that bed

 Suggested UI Flow

 Active Patients Tab:

 3438A — John Doe [Active]

 3438B — Mary Smith [Active]

 Archived Patients Tab:

-  3438A — John Doe (Discharged Jul 4)
-  2207B — Terry James (Discharged Jun 29)

Tapping into an archived patient shows their entire chart history (read-only).

Optional export/download from this screen.

#### Implementation Logic

Each note is stored with a:

room\_id

bed\_id

patient\_id or session ID

nurse\_id

timestamp

is\_archived flag (boolean)

On discharge:

Set is\_archived = true for the session/patient

Future charting under that room/bed generates a new patient session (even if you don't have EHR integration yet)

## Data Retention & Legal Notes (Canada)

PHIPA/HIPA in Canada often requires medical records to be retained for 10 years minimum

Even if you're not formally part of a hospital's EHR yet, you should never hard-delete chart data — just archive it securely

Later, when integrated with hospitals or clinics, you'll align with their internal retention policies

## Bonus Features (Phase 2+)

Add filters/search in archived view (by room, nurse, date range, patient initials)

Auto-archive after 24–48 hrs of patient being marked discharged

PDF summary auto-generated upon discharge for documentation/handover

## TL;DR

Don't delete anything — just archive it intelligently.

Keep current patients in an "Active" view and discharged patients in "Archived."

You'll stay compliant, organized, and future-proof.

Would you like a sample data schema or mock interface for this archive flow? I can mock it out or send JSON-style logic next.

QUESTION: okay thanks, also, I was wondering how we would save the client name assigned to a room in the initial stages, I assumed when a client is assigned via an external software, the changes reflect on our software, but since you said it's best not to focus on integration right now, how do you think it might work? after logging into the system and the room, you manually enter patient names and assign them to beds?

You're thinking in exactly the right direction again. Integration with external hospital systems (e.g. EHRs) would make this seamless eventually, but yes — for the MVP, you'll need a lightweight way to manually assign patient names to rooms/beds.

Let me give you a practical, low-tech but effective approach to handle patient assignment without EHR integration, so it still feels smooth and usable for nurses or admins.

#### MVP Plan: Manual Assignment with Session Logic

##### Goal:

Make it easy for a nurse (or shift lead) to assign a patient name or ID to a specific room and bed, and then chart notes under that assignment — all within the app, without touching external systems.

##### Suggested Flow for MVP:

###### Nurse or Admin Logs In

Lands on a "Room Overview" screen.

Sees something like:

###### Room 3438

[ Assign Patient to Bed A ]

[ Assign Patient to Bed B ]

##### They Click "Assign Patient" → Simple Modal:

Assign Patient to Room 3438B

 Name / ID: [ John Doe ]

 Admission Date: [ 2025-07-05 ]

 Bed: [ B ]

[ Save ]

Now, the system creates a "Patient Session" like:

```
{  
  "room": "3438",  
  "bed": "B",  
  "patient_name": "John Doe",  
  "admission_date": "2025-07-05",  
  "status": "active"  
}
```

 Then Charting Works Like This:

When the nurse holds to record voice → it's automatically tied to:

Room: 3438

Bed: B

Patient: John Doe

Nurse: angela123

Timestamped + stored

 Discharging the Patient

Later, someone can tap:

[ Discharge Patient ]

- Moves notes to "Archived"
- Room 3438B becomes available again

 Why This Works for MVP:

- No external integration needed
- Easy for 1–2 nurses to manage manually
- You get clean patient → bed → chart mapping
- Keeps your data tidy for AI summaries and exports
- You can simulate multiple patients easily during testing

 Bonus Ideas for Later

When you move past MVP, consider adding:

Optional "Patient Code" instead of full name (privacy)

Role-based access: Only admins can assign/discharge

Import from CSV for clinics that have digital records but no API

Notifications: "Room 3438B has no active patient — assign one to continue charting"

### Naming and Data Sensitivity Tip

Because you're not integrated yet, don't rely on official patient IDs unless you're sure you're compliant with data privacy laws. For now:

Store just a "display name" or alias ("Patient A", or initials)

Add a toggle for clinics to decide how much detail they want to input

### TL;DR

Until integration is possible, give nurses a manual but simple way to assign a patient to a room and bed, and automatically attach all voicecharted data to that "patient session".

It's safe, low-friction, and keeps your MVP clean and testable.

Let me know if you'd like a sample JSON schema for how to structure a PatientSession, or a quick UI sketch of the assign/discharge process.

QUESTION: Room 3438 \[ Assign Patient to Bed A \] \[ Assign Patient to Bed B \]

okay this all looks great, just a question, this is with the understanding that we have the data for each room of the clinic/hospital, including room numbers, beds in each room and so on right? that way I was thinking a nurse could log in with the room Id(if it exists in the clinic/hospital) and then see info like this: "Room 3438

[ Assign Patient to Bed A ]

[ Assign Patient to Bed B ]"

Yes — you're 100% right.  This whole workflow assumes you already have the basic room and bed data from the clinic or hospital you're piloting with.

That's key. Even if you're not integrated with their system (EHR/API), you can still ask for or build a static room/bed map at the start of the pilot. Here's how that works:

🔑 What You'll Need from the Clinic/Hospital (Pre-Integration)

Just ask for a basic room/bed structure like:

```
[  
 {  
   "room_number": "3438",  
   "beds": ["A", "B"]  
 },  
 {  
   "room_number": "2215",  
   "beds": ["A"]  
 },  
 {  
   "room_number": "1347",  
   "beds": ["A", "B", "C"]  
 }  
 ]
```

QUESTION: \*\*Goal:\*\* Add value beyond voice input,

do you feel there are other solid ways to do this as well, adding value beyond just voice input, to make us competitive

Absolutely — voice input is a great wedge, but it's not enough to hold market share once competitors catch on or larger EHR vendors improve their own voice features. You're smart to think ahead.

Here are 6 solid, defensible ways to add value beyond voice input, making Mobile Charter more competitive and sticky without requiring deep integrations yet:

 1. AI-Generated Shift Summaries

 "Give me a summary of everything I charted today."

At the end of a shift, generate a clean, human-readable summary of all their patient interactions.

Grouped by room → bed → time → status changes

Useful for:

Nurse handovers

Documentation review

Shift reports

 Competitive edge: No more scrolling through raw text or messy EHR timelines.

 2. Real-Time Smart Suggestions

 "Did you mean to say 'oriented x3'?" or "You haven't charted vitals for Room 1347A in 6 hours."

NLP layer listens for incomplete, vague, or critical charting gaps.

Flags things like:

Missing vitals

No documentation after meds

Unusual language or abbreviations

- Competitive edge: Adds clinical awareness, not just transcription.

 3. Mini Dashboard for Shift Coverage

 "Which patients have been charted today? Who's missing updates?"

Visual dashboard for each nurse or unit:

 Charted: Room 3438A (3 notes today)

 Room 2215B — No chart since 8 AM

Filters by time, activity level, or patient status

- Competitive edge: Transforms voice notes into actionable oversight.

 4. Audit-Ready Logging + Role-Based Views

 "Everything Angela charted this week — timestamped, verified."

Every chart note has:

Nurse ID

Time of input

Edit history

Role-based access:

Nurses see their own

Shift leads see whole unit

Admins can review archives

- Competitive edge: Helps facilities with compliance, dispute resolution, and training.

#### 5. Clinical Prompt Templates for Charting

 Tap a template: "Post-op patient check" → prompts nurse to record pain, vitals, wound condition

Reduce mental burden by offering reusable voice charting templates:

Pain assessment

Fall follow-up

Pre-discharge checks

- Competitive edge: Increases charting consistency + speed while reducing training gaps.

#### 6. Multilingual Support or Accent Adaptation

- Supports non-native English speakers with clarity and accuracy.

Train or fine-tune voice model to local accents, common phrases

Add support for voice-to-English from Tagalog, Hindi, French, etc.

- Competitive edge: Massive help in multicultural nursing environments.

- ✖ Optional Long-Term Ideas (Post-EHR Integration)

If you ever go deeper into systems integration:

Medication tracking ("Charted meds given 30 mins late")

Vitals trend visualization ("BP decreasing over 4 hrs")

AI-based patient condition alerts

Sync to physician dashboards or discharge planning tools

- TL;DR: How to Stay Competitive

Voice is your entry point — not your full moat. You stay ahead by offering:

Value Layer    Competitive Advantage

Summaries & Dashboards    Time-saving + clarity

Smart Prompts & Checks    Reduces errors

Templates    Speeds up routine workflows

Compliance Tools    Wins trust from admins

Accent/Language Support    Inclusivity = adoption

Would you like a feature prioritization matrix or user flow mockups for any of these ideas?  
That might help you decide which one to tackle next.