PRD

Product name (working)

BillCheck — Hospital bill sanity checker powered by CMS price transparency data

1. Problem statement

Patients receive hospital bills with line items and prices they do not understand and cannot evaluate. Although CMS requires hospitals to publish machine-readable price transparency files with negotiated rates, gross charges, and discounted cash prices, this data is inaccessible to patients in its raw form.
As a result, patients have no way to determine whether:

- a charge is typical,
- significantly higher than local norms, or
- potentially negotiable.

2. Target user

Primary user: Patient who has received a hospital facility bill (PDF) and wants to understand if the charges are reasonable.
Secondary user (future): Caregiver evaluating bills for a family member.

3. Goals (MVP)

The user can:

- Upload a hospital bill PDF
- See extracted line items (code, description, price)
- Confirm or correct those items
- Select the hospital that issued the bill
- See how each charge compares to that hospital's CMS-published prices and to other hospitals nearby
- Understand in plain language whether their bill is unusually high

4. Non-goals (explicitly out of scope for MVP)

- Professional/physician bills (facility only)
- Insurance EOB reconciliation
- Modifiers, units, revenue codes, bundling logic
- Perfect extraction accuracy from all PDFs
- User accounts or data persistence
- Appeals, letters, or negotiation tools (Phase 2)

5. Data sources

- CMS Hospital Price Transparency machine-readable files
- CMS "Shoppable Services" CPT/HCPCS list
- CMS Provider of Services file (hospital metadata)
- CPT/HCPCS/DRG public reference tables

6. User flow

Step 1 — Upload

User uploads a hospital bill PDF (or pastes text as fallback).
UI shows: File upload dropzone with loading state. Error messages displayed inline if processing fails.

Step 2 — Extraction review

System displays detected line items in an editable table format. Each row shows: code, description, quantity, and amount charged.
User can:

- Edit any field inline (description, code, price)
- Delete false positives
- Add missed items manually via "Add Missing Item" button

UI improvements: Amount field uses text input with currency formatting (dollar sign prefix, validates for positive values with max 2 decimal places). Labels clearly indicate "Amount Charged" for the price field. Extraction statistics shown above table (total items, comparable items, total charges).

Step 3 — Hospital confirmation

User selects hospital from a searchable list (autocomplete using CMS provider data). Shows hospital name, city, state, and approximate distance.

Step 4 — Results

For each line item:

- Your charge: $X
- Hospital's posted cash price: $Y
- Local range (50 miles): $A – $B
- Percentile ranking (with visual indicator)
- Simple explanation in plain language
- Expandable data provenance details

UI improvements: "Check Another Bill" button added to allow users to analyze multiple bills in same session. Visual percentile bars show where user's charge falls in local distribution. Color coding indicates if charge is below average (green), above average (amber), or significantly above average (red).

Step 5 — Bill summary

- Overall percentile of total charges
- Plain English interpretation
- Next steps recommendations

7. Functional requirements

7.1 PDF ingestion

- Accept PDF upload via drag-and-drop or file picker
- Extract text using pdfplumber library
- Show loading state during processing
- Display error messages if extraction fails
- If extraction fails, allow paste-in text as fallback

7.2 Line item detection

System attempts to detect:

- CPT / HCPCS / DRG codes via regex
- Service descriptions
- Associated prices
- Quantity (when present)

Returns extraction statistics: total items found, number comparable to CMS data, total charges.

7.3 User correction

User must be able to:

- Edit description
- Edit or add code
- Edit price (with proper currency validation)
- Delete false positives
- Add missed items manually

7.4 Hospital price ingestion

System stores normalized subset of hospital transparency data:

- Gross charge
- Discounted cash price
- Negotiated prices (optional for MVP)
- CPT/HCPCS/DRG mappings

Limited to a defined region or top N hospitals for MVP feasibility.

7.5 Comparison engine

For each line item:

- Find matching service in hospital file
- Compute local distribution across hospitals
- Compute percentile
- Identify price reference used

7.6 Results presentation

For each line item, display:

- User price vs hospital posted price
- Local min / max with visual range indicator
- Percentile with color-coded interpretation
- Data provenance link (expandable)
- "Why this might differ" explanation

8. UX requirements

- Mobile-first responsive design
- Clear, non-technical language
- Emphasis on explainability
- Visual percentile indicator (horizontal bar with color coding)
- Expandable "details" sections for data provenance
- Loading states for async operations
- Inline error handling with clear messaging
- Step indicators showing progress through flow
- "Check Another Bill" option on results screen

9. Data provenance (critical for trust)

Each comparison must show:

- Hospital source file name and date
- Which price field was used (gross charge, cash price, etc.)
- Confidence level of code match (high/medium/low)
- Number of hospitals in comparison
- Geographic radius used

10. Privacy & security posture (prototype)

- No data persistence
- PDFs processed in memory and discarded
- No PHI stored
- Clear disclaimer about prototype nature

11. Success metrics (for MVP validation)

- User can complete flow without instruction
- ≥80% of extracted line items require no correction (for test set)
- Users report understanding whether their bill is high or normal

12. Technical approach (prototype level)

- Frontend: React
- Backend: FastAPI Python server
- PDF parsing: pdfplumber library
- Storage: normalized subset of hospital price files (future)
- Code detection: regex + text heuristics
- No ML required for MVP

13. Known limitations (intentional)

- Some hospitals publish incomplete or inconsistent files
- CPT mapping may be imperfect when bills lack codes
- Approximately 40% of line items use hospital-internal codes not comparable to CMS data
- Prices may not reflect patient's exact insurance contract
- Facility vs professional billing differences

These are explained in the UI.

14. Phase 2 roadmap (documented, not built)

Epic 1 — Bill negotiation helper

- Generate appeal letter citing hospital's own data
- Provide call script

Epic 2 — Cheaper alternatives

- Show nearby hospitals with lower prices for same service
- Include quality indicators

Epic 3 — Extraction accuracy

- OCR for scanned PDFs
- Confidence scoring
- Learning from user corrections

Epic 4 — CPT code validation

- Validate codes against official CPT/HCPCS database
- Auto-populate descriptions for valid codes
- Flag invalid or unknown codes for user review

Epic 5 — Insurance-aware comparisons

- Payer and plan selection
- Negotiated rate relevance

Epic 6 — User accounts

- Save bill history
- Export reports
- Privacy controls