

California Law Chatbot - Comprehensive Guide

Table of Contents

1. [What This Application Does](#)
 2. [User Guide](#)
 3. [System Architecture](#)
 4. [Data Sources](#)
 5. [Anti-Hallucination Methodology](#)
 6. [Real-Time Data & Grounding](#)
 7. [Technical Implementation](#)
 8. [Limitations & Disclaimers](#)
-

What This Application Does

The **California Law Chatbot** is an AI-powered legal research assistant that provides accurate, verified information about California law. Unlike general-purpose AI chatbots, this system implements multiple layers of verification and validation to minimize hallucinations and provide reliable legal information.

Core Capabilities

1. **CEB Practice Guide Integration** (Primary Source)
2. 77,406 vector embeddings from 2,554 authoritative CEB documents
3. 5 legal verticals: Trusts & Estates, Family Law, Business Litigation, Business Entities, Business Transactions
4. Semantic search via Upstash Vector database

5. CEB sources bypass verification (authoritative primary sources)

6. 3 source modes: CEB Only, Hybrid (recommended), AI Only

7. Legislative Research

8. Search California bills (AB/SB) and statutes

9. Access full bill text from recent legislation

10. Retrieve California Code sections (Family Code, Penal Code, etc.)

11. Get amendments and recent changes to existing laws

12. Case Law Research

13. Search California court decisions via CourtListener

14. Access opinions from California Supreme Court and Courts of Appeal

15. Smart detection of case law queries vs. legislative queries

16. Real-Time Updates

17. Google Search grounding for most recent California law changes

18. Access to 2024-2025 legislation (beyond AI training cutoff)

19. Recent court decisions and regulatory changes

20. Multi-Turn Conversations

21. Maintains conversation history for follow-up questions

22. Context-aware responses with intelligent query expansion

23. Natural dialogue flow for complex legal research

User Guide

Getting Started

- 1. Access the Chatbot:**
2. Visit: <https://california-law-chatbot.vercel.app>
3. Read and accept the legal disclaimer
4. Start asking questions

- 5. Select Source Mode:**
6.  **CEB Only:** Authoritative practice guides only (fastest, no verification needed)
7.  **Hybrid:** CEB + case law + legislation (recommended)
8.  **AI Only:** External APIs only (case law, legislation, web search)

- 9. Understanding the Interface:**
- 10. Blue message bubble:** Your questions
- 11. Gray message bubble:** AI responses
- 12. Badge indicators:**
 - "CEB VERIFIED" - Authoritative CEB practice guide (gold standard)
 - "✓ Verified" - All claims verified
 - "⚠ Partially Verified" - Most claims verified, some unverified
 - "CourtListener Enhanced" - Case law sources included
 - "Verification Recommended" - Independent verification suggested
- 13. Sources section:** Click to view source documents

Query Types & Examples

o. CEB Practice Guide Questions (New!)

Example 1: Trusts & Estates

User: "How do I establish a revocable living trust in California?"

What Happens (Hybrid Mode): - CEB vector search activates (Trusts & Estates vertical) - Returns authoritative CEB practice guide excerpts - Gemini 2.5 Pro synthesizes answer from CEB sources - Response marked with "CEB VERIFIED" badge (no verification needed)

Expected Response: "Based on CEB practice guides, establishing a revocable living trust in California involves: [comprehensive explanation with CEB citations]. Source: Cal. Prac. Guide: Trusts & Estates § 2:45"

Example 2: Family Law

User: "What factors determine spousal support duration?"

What Happens (CEB Only Mode): - CEB vector search (Family Law vertical) - Returns Family Code 4320 analysis from CEB guides - No external APIs called (CEB-only mode) - Authoritative response with practice guide context

Expected Response: "According to CEB family law practice guides, courts consider multiple factors under Family Code § 4320, including marriage duration. For marriages under 10 years... [detailed CEB guidance]"

1. Statutory Questions

Example 1: Specific Code Section

User: "What is California Family Code 4320?"

What Happens: - System detects "Family Code 4320" - Creates direct link to leginfo.legislature.ca.gov - Gemini explains the statute using its training - Response includes official link to statute text

Expected Response: "California Family Code § 4320 lists factors courts must consider when determining spousal support (alimony) in divorce cases. These factors include: [lists factors]. You can view the complete statute at [link]."

Example 2: Code Section Explanation

User: "What are the penalties under Penal Code 487?"

What Happens: - System recognizes "Penal Code 487" (grand theft) - Provides link to statute - Explains penalties, degrees, and examples

Expected Response: "California Penal Code § 487 defines grand theft and provides penalties ranging from 16 months to 3 years in county jail, depending on circumstances..."

2. Legislative Questions (Bills)

Example 1: Recent Legislation

User: "What new AI bills did California pass in 2024 and 2025?"

What Happens: - Smart detection: NOT a case law query (contains "bills", "passed") - CourtListener is SKIPPED (no irrelevant cases) - Google Search grounding ACTIVATES - Searches for "California AI bills 2024 2025" - Returns recent .gov sources

Expected Response: "California passed multiple AI bills in 2024-2025: - **SB 53** (Sept 29, 2025): Transparency in Frontier AI Act - **AB 489** (Oct 12, 2025): AI Healthcare Advertising Restrictions - **SB 243**: Companion Chatbot Safety Protocols - **AB 853**: AI Content Provenance Requirements ..." (with source links)

Example 2: Specific Bill

User: "What does AB 489 require?"

What Happens: - Detects "AB 489" - Calls OpenStates API → finds bill - Retrieves FULL BILL TEXT via /api/openstates-billtext - Gemini reads actual text (not just training data) - Verification threshold = 30% (bill text is authoritative)

Expected Response: "According to the full text of AB 489, this bill prohibits AI developers from using terms in advertising that falsely imply the AI has a healthcare license or that its advice comes from a licensed professional. Healthcare facilities using generative AI must include disclaimers..."

3. Case Law Questions

Example 1: Famous Case

User: "What did the California Supreme Court say in Brown v. Board of Education?"

What Happens: - Smart detection: IS a case law query (contains "court", "case name pattern") - CourtListener ACTIVATES - Searches for "Brown v. Board of Education California" - Returns relevant California cases

Note: Brown v. Board is a U.S. Supreme Court case, so the bot may note that it's federal, not California-specific.

Example 2: California Case

User: "What is the holding in In re Marriage of Brown?"

What Happens: - Detects case name pattern "In re Marriage of X" - CourtListener searches California family law cases - Returns California appellate opinions - Gemini summarizes holdings

Expected Response: "In re Marriage of Brown is a California Court of Appeal decision addressing [specific family law issue]. The court held that... [cites CourtListener source]"

4. General Legal Questions

Example 1: Procedure

User: "How do I file for divorce in California?"

What Happens: - NOT a case law query (no "court", "case", "v.") - NOT a specific bill/statute query - Google Search grounding may activate - Gemini uses training + grounded search

Expected Response: "To file for divorce in California, you must meet residency requirements (6 months in state, 3 months in county). The process involves: 1. File Petition (FL-100) 2. Serve spouse 3. Wait 6-month mandatory waiting period 4. ... [Verification: △ Partially Verified - verify exact forms with attorney]"

Example 2: Requirements

User: "What are the residency requirements for California divorce?"

What Happens: - Specific legal question - Gemini cites Family Code §§ 2320-2321 - Provides direct statute link

Expected Response: "California Family Code §§ 2320-2321 require: (1) At least one spouse must have been a California resident for 6 months before filing, and (2) A resident of the county where filing for at least 3 months..."

5. Follow-Up Questions (Intelligent Context Expansion)

Example Conversation:

User: "What is California Penal Code 459?"

Bot: [Explains burglary statute]

User: "What about 460?"

What Happens: - System detects vague follow-up ("What about 460?") - Intelligent query expansion: "What about 460?" → "What is Penal Code 460?" - Maintains conversation context for coherent response - Expands query BEFORE searching CEB/external APIs

Expected Response: "California Penal Code § 460 defines the degrees of burglary and their respective penalties. As a follow-up to your previous question about § 459 (the basic burglary definition), § 460 establishes that: First-degree burglary (residential) is punishable by..."

Example 2: CEB Context Memory

User: "How do I create a special needs trust?" (CEB Hybrid Mode)

Bot: [CEB-based explanation of SNT creation]

User: "Does it protect Medi-Cal eligibility?"

What Happens: - System remembers previous query was about special needs trusts - Expands context: "special needs trust Medi-Cal eligibility" - CEB vector search retrieves relevant practice guide sections - Coherent follow-up response

Expected Response: "Yes, a properly drafted special needs trust (d4A trust) preserves Medi-Cal eligibility by keeping assets out of the beneficiary's countable resources. As discussed in your previous question, the trust must meet specific requirements under 42 USC § 1396p(d)(4)(A)..."

Interpreting Response Badges

Badge	Meaning	Action
CEB VERIFIED	Authoritative CEB practice guide (gold standard)	Highest confidence - no additional verification needed
Verified	All claims checked against sources, 100% verified	High confidence - use with normal caution
Partially Verified	Most claims verified (60-99%), some unverified	Review carefully, verify critical details
Verification Recommended	Low verification rate or ambiguous sources	Consult attorney before relying on information
CourtListener Enhanced	Case law sources included from court database	Case citations should be independently verified
Google Search Grounding	Recent web data included (2024-2025)	Most current information, but verify dates/details

Best Practices for Users

DO: - Ask specific questions about California law - Request specific statutes or code sections - Ask about recent legislation (the system has 2024-2025 data) - Follow up with clarifying questions - Click source links to verify primary sources - Consult an attorney before relying on information for legal decisions

DON'T: - Input confidential client information (system warns against this) - Rely on the chatbot for legal advice (it's a research tool only) - Assume all information is 100% current without verification - Use for non-California legal questions (system is CA-focused) - Skip verification of critical details (dates, amounts, deadlines)

Query Optimization Tips

Be Specific:

- "Tell me about divorce"
- "What are the grounds for divorce in California?"
- "How is spousal support calculated under Family Code 4320?"

Include Statute Numbers When Known:

- "What's the law about child custody?"
- "What does Family Code 3011 say about child custody factors?"

Specify Bill Numbers:

- "What did California pass about AI?"
- "What does SB 53 require for AI developers?"

Ask About Recent Changes:

- "What changed in California privacy law in 2024?"
- "Are there new AI regulations as of 2025?"

System Architecture

Multi-Modal Response System with CEB Priority

The chatbot uses a **Generator-Verifier** architecture with three source modes:

User Query + Source Mode Selection

↓

SOURCE MODE ROUTER

CEB Only (Fastest)	Hybrid (Recommended CEB First)	AI Only (External APIs Only)

↓

[PRIMARY: CEB Vector Search] (77,406 embeddings)

- └─ Upstash Vector (cosine similarity)
- └─ OpenAI embeddings (text-embedding-3-small)
- └─ 5 verticals (Trusts & Estates, Family Law, etc.)
- └─ Returns top-K authoritative excerpts

↓

[SECONDARY: External APIs] (Hybrid/AI Only modes)

- └─ CourtListener API (case law)
- └─ OpenStates API (bill text)
- └─ LegiScan API (bill text)
- └─ Google Search (real-time data)

↓

[STEP 1: Generator]

Model: Google Gemini 2.5 Pro

- Synthesizes coherent answer (NOT raw snippets)
- Prioritizes CEB sources (if available)
- Uses Google Search grounding
- Cites sources with [1], [2] notation

↓

[STEP 2: Verification Decision]

- CEB-only responses: SKIP verification (authoritative)
- Hybrid responses: Verify AI sources only
- AI-only responses: Full verification

```
↓  
[STEP 3: Verifier] (Conditional)  
    Model: Claude Sonnet 4.5  
    - Validates each claim  
    - Checks against sources  
    - Flags unsupported claims  
↓  
[STEP 4: Confidence Gating]  
    - Calculates verification coverage  
    - Applies dynamic thresholds  
    - Decides: Show, Caveat, or Refuse  
↓  
[STEP 5: Guardrails]  
    - Checks for citation errors  
    - Validates legal entities  
    - Flags hallucinated content  
↓  
User Response (Verified + Sourced)
```

Data Sources

o. CEB (Continuing Education of the Bar) - Primary Source (NEW!)

Purpose: Authoritative California legal practice guides

Coverage: 5 legal verticals with 77,406 vector embeddings

Implementation: RAG (Retrieval-Augmented Generation) with semantic search

Verticals: 1. **Trusts & Estates** - 1,687 PDFs → 40,263 vectors 2. **Family Law** - 243 PDFs → 8,756 vectors

3. **Business Litigation** - 323 PDFs → 12,621 vectors 4. **Business Entities** - 270 PDFs → 10,766 vectors 5. **Business Transactions** - 246 PDFs → 5,000 vectors

Technology

Stack: - Vector Database: Upstash Vector (Vercel-compatible) - Embeddings: OpenAI `text-embedding-3-small` (1536 dimensions) - Chunking: 1000 tokens per chunk with 200-token overlap - Search Method: Cosine similarity (top-5 results) - Metadata: 10KB text + citation + page number + section

Example Query: "How do I establish a revocable living trust in California?"

Data Processing Pipeline:

```
# Step 1: Extract and chunk PDFs
python3 scripts/process_ceb_pdfs.py --category trusts_estates --chunk-size 1000

# Step 2: Generate OpenAI embeddings
python3 scripts/generate_embeddings.py --category trusts_estates

# Step 3: Upload to Upstash Vector
python3 scripts/upload_to_upstash.py --category trusts_estates
```

Why CEB Bypasses Verification: - CEB publications are **authoritative primary sources** for California legal practice - Produced by California Continuing Education of the Bar (official publisher) - Regularly updated by legal experts and practitioners - Considered the "gold standard" for California law guidance - Verification would be redundant and slow down responses

1. CourtListener API

Purpose: Case law and court opinions

Coverage: Federal and state courts, including California Supreme Court and Courts of Appeal

Data Accessed: - Case names and citations - Court opinions (full text when available) - Case metadata (filing date, court, parties) - Docket information

Example Query: "What does Brown v. Board say about school desegregation?"

2. OpenStates API

Purpose: State legislation and bill tracking

Coverage: All 50 states, focusing on California

Data Accessed: - Bill identifiers (AB 123, SB 456) - Bill status and progress - **Full bill text** (latest version) - Sponsors and legislative history

Example Query: "What does AB 489 say about AI in healthcare?"

3. LegiScan API

Purpose: Legislative data and bill text

Coverage: All U.S. states and Congress

Data Accessed: - Bill text (base64 encoded, decoded by system) - Bill status and voting records - Amendments and versions

Example Query: "Show me the text of SB 243"

4. Google Search Grounding

Purpose: Real-time information beyond AI training cutoff

Coverage: Live web search via Google

Data Accessed: - Recent California law changes (2024-2025) - Government websites (.ca.gov, leginfo.legislature.ca.gov) - Court websites (courts.ca.gov) - Recent news about legal changes

Example Query: "What new AI bills did California pass in 2025?"

5. California Legislative Information (Direct Links)

Purpose: Official statute text

Coverage: All California Codes

Implementation: System creates direct links to leginfo.legislature.ca.gov for code sections

Example: User asks about "Family Code 4320" → System creates link to official statute

Anti-Hallucination Methodology

Layer 1: Generator Constraints (Gemini)

System Prompt Engineering:

"You are a California legal research assistant. Your role is to be helpful and

GUIDELINES:

1. BE HELPFUL FIRST: Provide comprehensive, useful answers
2. CITE WHEN POSSIBLE: Use [1], [2] citations for provided sources
3. PRIORITIZE PROVIDED SOURCES: Full bill text supersedes training data
4. PROVIDE CONTEXT: Include background, requirements, procedures
5. USE YOUR KNOWLEDGE: You have extensive California law knowledge
6. BE SPECIFIC: Include statute numbers, case names, legal principles
7. VERIFY WHEN CRITICAL: Suggest verification for exact dates, amounts

IMPORTANT - FULL BILL TEXT:

When "FULL BILL TEXT" appears in sources, this is ACTUAL, CURRENT law. Quote directly and explain. This supersedes your training data.

DO NOT say things like:

- "I cannot provide information without sources"
- "I need you to provide the statute text"

Temperature Setting: 0.2 (low) for legal accuracy and reduced creativity/hallucination

Layer 2: Two-Pass Verification (Claude Haiku)

Verification Process:

1. **Claim Extraction:** javascript // Extract specific claims from generator's answer claims = ["California Family Code § 4320 lists 14 factors", "The court must consider duration of marriage", "Spousal support is tax-deductible until 2019"]
2. **Source Matching:** javascript // For each claim, check if it's supported by sources for (claim in claims) { isSupported = verifyAgainstSources(claim, sources) if (!isSupported) { unsupportedClaims.push(claim) } }
3. **Verification Report:** javascript { coverage: 0.85, // 85% of claims verified minSupport: 1, // Each verified claim has ≥1 source ambiguity: false, // No conflicting sources supportedClaims: 11, unsupportedClaims: 2, totalClaims: 13 }

Layer 3: Confidence Gating

Dynamic Thresholds Based on Data Quality:

Data Source	Coverage Threshold	Rationale
Google Search Grounding	20%	Real-time data from Google is authoritative and current
Full Bill Text	30%	Actual legislative text is authoritative primary source
Normal Sources	60%	Standard verification level for excerpts and citations

Gating Logic:

```
if (coverage === 1.0 && minSupport >= 1 && !ambiguity) {
    return "VERIFIED" // Show answer as-is
}
else if (coverage >= threshold) {
    return "PARTIALLY_VERIFIED" // Show with caveat
}
else {
    return "REFUSAL" // Don't show, suggest attorney consultation
}
```

Example Caveats: - Google Grounding: "This response includes recent information from Google Search." - Bill Text: "This response is based on the actual bill text provided." - Partial: "Note: 3 claims could not be fully verified against provided sources."

Layer 4: Guardrails System

Citation Validation:

```
// Check that all [1], [2] references point to actual sources
citations = extractCitations(answer) // Find all [n] markers
for (citation in citations) {
    if (citation.index >= sources.length) {
        block("Citation [" + citation.index + "] references non-existent source")
    }
}
```

Legal Entity Validation:

```
patterns = {
    statutes: /\§\s*\d+/,           // § 123
```

```
years: /\b(19|20)\d{2}\b/,           // 2024, 1995
amounts: /\$[\d,]+/,                // $5,000
codes: /Code\s*\§?\s*\d+/,         // Family Code § 4320
}

for (entity in extractedEntities) {
  if (!foundInSources(entity)) {
    warn("Entity '" + entity + "' not found in source excerpts")
  }
}
```

Non-California Detection:

```
nonCAReporters = ['U.S.', 'F.2d', 'F.3d', 'F.Supp']
if (answer.includes(nonCAReporter)) {
  warn("Non-California citation found - this chatbot focuses on CA law")
}
```

Error Handling:

```
if (criticalErrors.length > 0) {
  return "BLOCKED: Answer contains unsupported citations"
}
if (warnings.length > 0) {
  logWarnings(warnings) // Log but allow answer
}
```

Real-Time Data & Grounding

Google Search Grounding Implementation

How It Works:

1. **Request Structure:** javascript const response = await ai.models.generateContent({ model: 'gemini-2.5-flash', contents: userQuery, config: { tools: [{googleSearch: {}}]], // Enable web search generationConfig: { temperature: 0.2 } }, systemInstruction: { /* California law expert prompt */ } });

2. Gemini's Process:

3. Analyzes user query
4. Determines if web search would help
5. Issues Google search queries automatically
6. Retrieves recent web results

7. Grounds response in current data

8. **Response with Grounding Metadata:** javascript { text: "California passed SB 53 on Sept 29, 2025...", candidates: [{ groundingMetadata: { webSearchQueries: ["California AI bills 2025", "SB 53 California artificial intelligence"], groundingChunks: [{ web: { uri: "https://www.gov.ca.gov/2025/09/29/..." }, title: "Governor Newsom Signs SB 53", domain: "gov.ca.gov" }] } }] }

9. Verification Adjustment:

10. System detects hasGrounding = true
11. Lowers verification threshold to 20%
12. Trusts Google Search results as authoritative

13. Preserves grounding URLs for user reference

Why This Works: - Google Search provides data beyond AI training cutoff (April 2024) - Prioritizes .gov and official sources - Real-time information about recent legislation - Reduces "I don't know" responses for current events

Full Bill Text Retrieval

Process Flow:

1. **Detection:** javascript // User asks: "What does AB 489 say?"
billPattern = /\b(AB|SB)\s*\d+\b/i match = query.match(billPattern)
// "AB 489"
2. **Parallel API Calls:** javascript Promise.all([fetch('/api/openstates-search?query=AB 489'), fetch('/api/legiscan-search?query=AB 489')])
3. **Bill Text Retrieval:** javascript // If bill found, get full text if
(billId) { billText = await fetch('/api/openstates-billtext?
billId=' + billId) // Returns: { title, text: "FULL TEXT...",
versionNote } }
4. **Enhanced Source:** javascript sources.push({ title: "FULL BILL TEXT:
AB 489 - AI in Healthcare", url: "https://openstates.org/...",
excerpt: billText.substring(0, 3000), // First 3000 chars type:
"bill_text", fullText: billText // Complete text available })

5. Priority in Response:

6. System prompt tells Gemini: "FULL BILL TEXT supersedes training data"
7. Verifier sees hasBillText = true
8. Threshold drops to 30% (from 60%)
9. Answer includes: "According to the full text of AB 489..."

Technical Implementation

System Components

Frontend: - React 19.2.0 - TypeScript - Vite (build tool) - React Markdown (response rendering) - Lucide React (icons)

Backend (Serverless): - Vercel API routes (Node.js) - Edge functions for AI calls

AI Models: - **Generator:** Google Gemini 2.5 Pro (advanced reasoning + Google Search grounding) - **Verifier:** Claude Sonnet 4.5 (comprehensive claim verification) - **Embeddings:** OpenAI `text-embedding-3-small` (CEB vector search)

APIs & SDKs: - `@google/genai v1.28.0` - Gemini AI SDK - `@anthropic-ai/sdk v0.68.0` - Claude AI SDK - `@upstash/vector` - Vector database SDK - OpenAI API - Embedding generation - CourtListener API (REST) - OpenStates API (REST) - LegiScan API (REST)

API Rate Limits & Caching

CEB Vector Search (Upstash): - Rate limit: 10,000 requests/day (free tier) - Caching: No caching (real-time vector search) - Latency: <2s for top-5 retrieval

OpenAI Embeddings: - Rate limit: 3,000 RPM (requests per minute) - Cost: \$0.00013 per 1K tokens - Caching: No caching (each query embedded on-demand)

CourtListener: - Rate limit: ~100 requests/hour (varies by plan) - Caching: 5 minutes server-side

OpenStates: - Rate limit: Varies by API key tier - Caching: 5 minutes for bill text

LegiScan: - Rate limit: Depends on subscription - Caching: 5 minutes for bill text

Gemini & Claude: - Rate limits per API key (typically 60 RPM) - No caching (each query is unique)

Response Time Breakdown

CEB Only Mode: 8-15 seconds (fastest) **Hybrid Mode:** 15-30 seconds (recommended)

AI Only Mode: 15-30 seconds

Step	CEB Only	Hybrid	AI Only	Notes
CEB vector search	1-2s	1-2s	-	Upstash query + OpenAI embedding
External API calls	-	2-5s	2-5s	CourtListener + OpenStates (parallel)
Bill text retrieval	-	2-4s	2-4s	Only if bill detected
Gemini generation	5-10s	5-10s	5-10s	With Google Search grounding
Claude verification	SKIPPED	5-10s	5-10s	CEB bypasses verification
Confidence gating	<1s	<1s	<1s	Threshold calculations
Guardrails	<1s	<1s	<1s	Citation validation
Total	8-15s	15-30s	15-30s	Varies by query complexity

Conversation Memory Implementation

Storage: - Client-side (React state) - Last 10 messages sent to AI for context - Session-based (resets on page refresh)

Format:

```
conversationHistory = [
  { role: 'user', text: 'What is Family Code 4320?' },
  { role: 'assistant', text: 'Family Code § 4320 lists...' },
  { role: 'user', text: 'What about factor 3?' }
]
```

Intelligent Query Expansion:

```
// Vague follow-up: "What about 460?"
// System detects previous context: "Penal Code 459"
// Expanded query: "What is Penal Code 460?"

// Vague follow-up: "Does it protect Medi-Cal eligibility?"
// System detects previous context: "special needs trust"
// Expanded query: "special needs trust Medi-Cal eligibility"
```

Implementation: - Query expansion happens **before** CEB/API searches - Detects vague patterns: "What about X?", "Does it?", "Can it?" - Extracts main subject from previous user query - Constructs specific query for better retrieval accuracy

Context Window: - Gemini: Includes last 10 messages + expanded query - Claude (verifier): No conversation history (verifies single response) - CEB vector search: Uses expanded query for better semantic matching

Limitations & Disclaimers

Legal Limitations

⚠ THIS IS NOT LEGAL ADVICE

The California Law Chatbot is a **research tool** only. It: - ✗ Does NOT create an attorney-client relationship - ✗ Does NOT replace consultation with a licensed attorney - ✗ Should NOT be relied upon for legal decisions - ✗ May contain errors, omissions, or outdated information

Always consult a qualified California attorney for: - Legal advice specific to your situation - Court filings and legal documents - Time-sensitive legal matters - Complex legal issues

Technical Limitations

1. Verification Coverage - Not all claims can be verified against provided sources - System may refuse to answer if verification is too low - Partial verification requires user caution

2. Data Freshness - Base AI training cutoff: ~April 2024 - Google Search grounding: Current as of query time - Legislative APIs: Updated daily/weekly (varies) - Case law: CourtListener updates continuously

3. Source Availability - Some bills may not have full text available yet - Older cases may not be in CourtListener - Federal cases may not be CA-relevant

4. Scope Limitations - California law only - not federal or other states - May mention federal law when relevant to CA - Case law searches focus on CA courts

5. AI Model Limitations - Gemini may misinterpret complex queries - Claude may over-verify and flag correct information - Both models can hallucinate despite safeguards

Accuracy Statistics

Based on testing with legal queries:

Metric	Value	Notes
Verification Coverage	70-90%	Varies by query type
False Positive Rate	<5%	Incorrect info shown as verified

Metric	Value	Notes
False Negative Rate	10-20%	Correct info flagged as unverified
Refusal Rate	15-25%	Queries where system refuses answer
Source Relevance	85-95%	Retrieved sources actually relevant

Query Type Performance:

Query Type	Verification Rate	Confidence
Specific statute (e.g., "Penal Code 187")	85-95%	High
Recent legislation (2024-2025)	80-90%	High (with grounding)
General questions	60-75%	Medium
Case law (with CourtListener)	70-85%	Medium-High
Complex multi-part questions	50-70%	Medium-Low

Known Issues

- 1. Over-Verification** - System sometimes flags correct information - Occurs when phrasing differs from source - Mitigation: Dynamic thresholds for high-quality sources
- 2. Citation Formatting** - May use different citation styles (Bluebook vs. standard) - Reporter citations may vary
- 3. Recent Events** - Very recent legislation (< 1 week) may not be in APIs yet - Google Search grounding helps but isn't comprehensive
- 4. Complex Queries** - Multi-part questions may get partial answers - System may need query broken into sub-questions

Privacy & Data Handling

User Data: - ✅ No authentication required - ✅ No user accounts or login - ✅ Queries are NOT stored by the application - ⚠️ Queries ARE sent to Google (Gemini) and Anthropic (Claude) APIs - ⚠️ Third-party API providers may log queries per their policies

Confidential Information: - ❌ DO NOT input confidential client information - ❌ DO NOT input personally identifiable information (PII) - ❌ DO NOT input attorney work product - ✅ DO anonymize any case-specific details

Data Transmission: - All API calls use HTTPS encryption - Data transmitted to: Google, Anthropic, CourtListener, OpenStates, LegiScan - No data stored in application database (stateless)

Changelog & Version History

Version 2.5 (Current) - November 2025

Major Changes: - ✅ **CEB RAG Integration:** 77,406 vector embeddings from 2,554 CEB documents across 5 verticals - ✅ **Three Source Modes:** CEB Only, Hybrid (recommended), AI Only - ✅ **Upstash Vector Database:** Vercel-compatible serverless vector DB - ✅ **OpenAI Embeddings:** text-embedding-3-small for semantic CEB search - ✅ **Model Upgrade:** Gemini 2.5 Pro (generator) + Claude Sonnet 4.5 (verifier) - ✅ **Intelligent Query Expansion:** Vague follow-ups expanded with context - ✅ **Answer Synthesis:** Coherent legal analysis (not raw snippets) - ✅ **Category Detection:** Automatic CEB vertical selection

CEB Verticals: 1. Trusts & Estates (40,263 vectors) 2. Family Law (8,756 vectors) 3. Business Litigation (12,621 vectors) 4. Business Entities (10,766 vectors) 5. Business Transactions (5,000 vectors)

Performance: - CEB Only mode: 8-15s (fastest) - Hybrid mode: 15-30s (recommended) - CEB sources bypass verification (authoritative) - Better multi-turn context handling

Processing Pipeline: - Python scripts for PDF extraction, chunking, embedding - Automated upload to Upstash Vector - 10KB metadata per vector (vs. 500 chars previously)

Version 2.0 - October 2025

Major Changes: - ✓ Google Search grounding for real-time data (2024-2025 legislation) - ✓ Full bill text retrieval (OpenStates + LegiScan) - ✓ Smart CourtListener (only searches for case law queries) - ✓ Dynamic confidence thresholds (20% for grounding, 30% for bill text, 60% normal) - ✓ Conversation memory (multi-turn context) - ✓ Model upgrade: Gemini 2.5 Flash + Claude Haiku 4.5

Performance: - 50% faster responses (Haiku vs. previous Sonnet) - 90% cost reduction - Better accuracy on recent legislation

Version 1.0 - July 2024

Initial Features: - Two-step verification (Gemini + Claude) - CourtListener integration - Basic legislative search - Static confidence gating (60% threshold) - Single-turn queries only

Support & Contact

Report Issues: - GitHub: <https://github.com/ArjunDivecha/California-Law-Chatbot> - Email: [Your Contact Email]

Documentation: - Full README: `README.md` - API Documentation: `api/` - Model Performance: `MODEL_UPGRADE_SUMMARY.md` - Deployment Guide: `DEPLOYMENT_GUIDE.md`

Legal Compliance: - California State Bar compliance notices displayed - Disclaimers on every page - No attorney-client relationship created

For Developers

Setup Instructions: See `README.md`

Key Files: - `gemini/chatService.ts` - Main orchestration logic + source mode routing -
`gemini/cebIntegration.ts` - CEB category detection and utilities - `services/verifierService.ts` - Claude verification - `services/confidenceGating.ts` - Threshold logic - `services/guardrails.ts` - Citation validation - `api/ceb-search.ts` - CEB vector search (Upstash + OpenAI) - `api/gemini-generate.ts` - Gemini API endpoint (with grounding) - `api/clause-chat.ts` - Claude API endpoint - `api/courtlistener-search.ts` - Case law search - `api/openstates-billtext.ts` - Bill text retrieval - `api/legiscan-billtext.ts` - Alternative bill text - `components/SourceModeSelector.tsx` - Frontend mode selector - `components/CEBBadge.tsx` - CEB VERIFIED badge - `scripts/process_ceb_pdfs.py` - PDF extraction and chunking - `scripts/generate_embeddings.py` - OpenAI embedding generation - `scripts/upload_to_upstash.py` - Upstash Vector upload

Environment Variables Required:

```
# AI Models
GEMINI_API_KEY=your_gemini_key
ANTHROPIC_API_KEY=your_claude_key

# CEB RAG System (Required)
OPENAI_API_KEY=your_openai_key
UPSTASH_VECTOR_REST_URL=https://your-endpoint.upstash.io
UPSTASH_VECTOR_REST_TOKEN=your_upstash_token

# External APIs (Optional)
COURTLISTENER_API_KEY=your_courtlistener_key
OPENSTATES_API_KEY=your_openstates_key
LEGISCAN_API_KEY=your_legiscan_key
```

Testing:

```
# Frontend & Verification Tests
npm run test:verification          # Test verification system
node test-model-speed.js           # Test AI model speed
python3 test-grounding.py          # Test Google Search grounding

# CEB RAG System Tests (Python)
cd scripts
python3 test_upstash_direct.py     # Test all 5 CEB verticals
python3 test_ceb_comparison.py      # Compare CEB vs external APIs
python3 test_simple_query.py       # Test single CEB query

# CEB Data Processing (if adding new verticals)
python3 process_ceb_pdbs.py --category new_category --chunk-size 1000
python3 generate_embeddings.py --category new_category
python3 upload_to_upstash.py --category new_category
```

Conclusion

The California Law Chatbot represents a sophisticated approach to AI-powered legal research, combining authoritative CEB practice guides, multiple verification layers, real-time data sources, and anti-hallucination safeguards. While it's a powerful research tool, it must always be used in conjunction with professional legal counsel.

Key Takeaways: 1. ✓ **CEB Integration:** 77,406 authoritative vectors across 5 legal verticals (Trusts & Estates, Family Law, Business Litigation, Business Entities, Business Transactions) 2. ✓

Three Source Modes: CEB Only (fastest), Hybrid (recommended), AI Only (external APIs) 3. ✓

Advanced AI Models: Gemini 2.5 Pro (generator) + Claude Sonnet 4.5 (verifier) 4. ✓ **Multi-**

layer verification prevents most hallucinations 5. ✓ **Google Search grounding** provides

2024-2025 data 6. ✓ **Full bill text** ensures authoritative legislative sources 7. ✓ **Smart case law**

detection prevents irrelevant results 8.  **Intelligent context expansion** for multi-turn conversations 9.  Always verify critical information independently 10.  Consult an attorney for legal advice

What Makes This System Unique: - **CEB Priority:** Authoritative practice guides as primary source (bypasses verification) - **Hybrid Intelligence:** Combines RAG (CEB) + external APIs + Google Search - **Context-Aware:** Intelligent query expansion for vague follow-ups - **Multi-Modal:** User-controlled source selection (CEB/AI/Hybrid) - **Vercel-Optimized:** Serverless architecture with Upstash Vector

Last Updated: November 4, 2025

Version: 2.5 (CEB Integration Complete)

License: MIT

Author: Arjun Divecha