

California Law Chatbot

Comprehensive Guide

October 2025

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. Legislative Research**

2. Search California bills (AB/SB) and statutes
3. Access full bill text from recent legislation
4. Retrieve California Code sections (Family Code, Penal Code, etc.)
5. Get amendments and recent changes to existing laws

6. Case Law Research

7. Search California court decisions via CourtListener
8. Access opinions from California Supreme Court and Courts of Appeal
9. Smart detection of case law queries vs. legislative queries

10. Real-Time Updates

11. Google Search grounding for most recent California law changes
12. Access to 2024-2025 legislation (beyond AI training cutoff)
13. Recent court decisions and regulatory changes

14. Multi-Turn Conversations

15. Maintains conversation history for follow-up questions
 16. Context-aware responses based on previous queries
 17. 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. Understanding the Interface:

6. Blue message bubble: Your questions

7. Gray message bubble: AI responses

8. Badge indicators:

- "✓ Verified" - All claims verified
- "⚠ Partially Verified" - Most claims verified, some unverified
- "CourtListener Enhanced" - Case law sources included
- "Verification Recommended" - Independent verification suggested

9. Sources section: Click to view source documents

Query Types & Examples

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: Δ 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 (Conversation Memory)

Example Conversation:

User: "What is California Family Code 4320?"

Bot: [Explains spousal support factors]

User: "What about the duration of marriage factor?"

What Happens: - System maintains conversation history - Knows "the duration of marriage factor" refers to Family Code 4320 - Provides context-aware answer

Expected Response: "Regarding Family Code § 4320's duration of marriage factor from your previous question: California courts consider how long the couple was married. For marriages under 10 years ('short-term'), there's typically a presumption that support lasts half the length of the marriage. For marriages over 10 years ('long-term'), courts retain jurisdiction indefinitely..."

Interpreting Response Badges

Badge	Meaning	Action
✓ 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

Two-Step Verification System

The chatbot uses a **Generator-Verifier** architecture to ensure accuracy:

User Query

↓

[External Data Sources]

- ├ 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 Flash

- Generates answer with claims
- Uses Google Search grounding
- Cites provided sources

↓

[STEP 2: Verifier]

Model: Claude Haiku 4.5

- Validates each claim
- Checks against sources
- Flags unsupported claims

↓

[STEP 3: Confidence Gating]

- Calculates verification coverage
- Applies dynamic thresholds
- Decides: Show, Caveat, or Refuse

↓

[STEP 4: Guardrails]

- Checks for citation errors
- Validates legal entities
- Flags hallucinated content

↓

User Response (Verified)

Data Sources

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 informative."

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 Flash (speed + quality) - **Verifier:** Claude Haiku 4.5 (speed + accuracy)

APIs & SDKs: - @google/genai v1.28.0 - Gemini AI SDK - @anthropic-ai/sdk v0.68.0 - Claude AI SDK - CourtListener API (REST) - OpenStates API (REST) - LegiScan API (REST)

API Rate Limits & Caching

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

Typical query: **15-30 seconds**

Step	Time	Notes
API calls (parallel)	2-5s	CourtListener + OpenStates/LegiScan
Bill text retrieval	2-4s	Only if bill detected
Gemini generation	5-10s	With Google Search grounding
Claude verification	5-10s	Two-pass claim checking
Confidence gating	<1s	Threshold calculations
Guardrails	<1s	Citation validation
Total	15-30s	Varies by query complexity

Conversation Memory Implementation

Storage: - Client-side (React state) - Last 10 messages sent to AI for context

Format:

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

Context Window: - Gemini: Includes last 10 messages - Claude (verifier): No conversation history (verifies single response)

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
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 2.0 (Current) - 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 - `services/verifierService.ts` - Claude verification - `services/confidenceGating.ts` - Threshold logic
- `services/guardrails.ts` - Citation validation - `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

Environment Variables Required:

```
GEMINI_API_KEY=your_gemini_key
ANTHROPIC_API_KEY=your_claude_key
COURTLISTENER_API_KEY=your_courtlistener_key
OPENSTATES_API_KEY=your_openstates_key
LEGISCAN_API_KEY=your_legiscan_key
```

Testing:

```
# Run verification system test
npm run test:verification

# Run model speed test
node test-model-speed.js

# Run grounding test
python3 test-grounding.py
```

Conclusion

The California Law Chatbot represents a sophisticated approach to AI-powered legal research, combining 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. ✓ Multi-layer verification prevents most hallucinations 2. ✓ Google Search grounding provides 2024-2025 data 3. ✓ Full bill text ensures authoritative legislative sources 4. ✓ Smart case law detection prevents irrelevant results 5. ⚠ Always verify critical information independently 6. ⚠ Consult an attorney for legal advice

Last Updated: October 30, 2025

Version: 2.0

License: MIT

Author: Arjun Divecha