



Judicature MVP Development Plan

Comprehensive Legal Technology Platform for India

Based on your product requirements and regulatory analysis, here's a detailed MVP plan that prioritizes compliance, user value, and rapid market entry while navigating BCI Rule 36 restrictions and DPDP Act requirements.

Executive Summary

Judicature will be built as a three-phase MVP targeting clients, lawyers, and legal researchers with core focus on:

- **Compliance-first design** respecting BCI advertising restrictions
- **DPDP-compliant data handling** with explicit consent mechanisms
- **NJDG integration** for comprehensive judgment access
- **AI-powered legal assistance** with clear disclaimers
- **Secure communication** with end-to-end encryption

MVP Phase Structure

Phase 1 (Weeks 1-8): Foundation & Core Features

Timeline: 2 months | **Budget:** ₹15-25 lakhs | **Team:** 4-6 developers

Core Features:

- Multi-role authentication system (Client/Lawyer/Admin)
- Lawyer verification and neutral directory
- Basic case intake and management
- Secure messaging with E2E encryption
- NJDG search integration MVP
- AI legal assistant with disclaimers
- DPDP consent management

Phase 2 (Weeks 9-16): Enhanced Workflows

Timeline: 2 months | **Budget:** ₹20-30 lakhs | **Team:** 6-8 developers

Enhanced Features:

- Advanced case document workflows
- Court hearing management
- Payment processing integration
- Mobile web optimization
- Analytics dashboard MVP
- Advanced search filters

Phase 3 (Weeks 17-24): Scale & Intelligence

Timeline: 2 months | **Budget:** ₹25-35 lakhs | **Team:** 8-10 developers

Advanced Features:

- Predictive case analytics
- Native mobile apps (iOS/Android)
- Enterprise lawyer dashboard
- API marketplace for integrations
- Advanced AI legal research

Core Features Breakdown

1. Authentication & User Management

Compliance Focus: DPDP consent capture, KYC verification

Features:

- Multi-factor authentication (Email/SMS OTP)
- Role-based access control (Client/Lawyer/Researcher/Admin)
- KYC document verification workflow
- DPDP-compliant consent management
- Profile management with audit trails

Technical Stack:

- JWT tokens with refresh mechanism
- Aadhaar/PAN verification APIs
- Document OCR for KYC automation

- Audit logging for all profile changes

2. Lawyer Discovery & Matching

Compliance Focus: BCI Rule 36 - No advertising/promotional content

Features:

- Neutral lawyer directory (no rankings/promotions)
- Practice area filtering without "specialization claims"
- Location-based matching
- Availability status
- Verification badges only
- Experience display (years, not expertise claims)

Technical Implementation:

- Geospatial search using PostGIS
- Tag-based filtering system
- Verification workflow with manual review
- Compliance monitoring for promotional content

3. Secure Communication Hub

Compliance Focus: Attorney-client privilege protection

Features:

- End-to-end encrypted messaging
- File sharing with virus scanning
- Video/voice call integration
- Message retention policies
- Export functionality with client consent
- Read receipts and typing indicators

Technical Stack:

- Signal Protocol for E2E encryption
- WebRTC for video calls
- AWS S3/Azure Blob for file storage
- Message queuing with Redis

4. Case Management System

Features:

- Case intake wizard with guided forms
- Matter lifecycle tracking
- Task management and deadlines
- Court hearing calendar
- Document repository with version control
- Time tracking and billing (optional)
- Status updates and notifications

Database Schema:

```
Cases: id, client_id, lawyer_id, case_type, status, court_jurisdiction
Tasks: id, case_id, title, due_date, status, assigned_to
Documents: id, case_id, filename, file_path, uploaded_by, timestamp
Hearings: id, case_id, court_name, hearing_date, status, notes
```

5. NJDG Integration & Judgment Search

Features:

- Real-time judgment search across all Indian courts
- Advanced filtering (court, date, party names, citation)
- Bookmark and save functionality
- Citation tracking and cross-references
- PDF download and sharing
- Search history and analytics

Technical Implementation:

- NJDG API integration with rate limiting
- Elasticsearch for judgment indexing
- OCR processing for scanned judgments
- Metadata extraction and tagging
- Full-text search with relevance scoring

6. AI Legal Assistant

Compliance Focus: Clear disclaimers - "Not legal advice"

Features:

- Multi-language query processing (Hindi/English)

- Case law similarity matching
- Confidence scoring for responses
- Citation of relevant judgments
- Escalation to human lawyers
- Query history and analytics

Technical Stack:

- Large Language Model (GPT-4/Claude) with legal training
- Vector database for judgment embeddings
- RAG (Retrieval Augmented Generation) architecture
- Confidence threshold for response quality

7. DPDP Compliance Module

Features:

- Granular consent collection
- Data subject rights (DSR) portal
- Consent withdrawal mechanisms
- Data breach notification system
- Privacy policy management
- Cross-border transfer controls

Dashboard Designs

Client Dashboard

Main Sections:

- **My Cases:** Active matters, status updates, next steps
- **Find Lawyers:** Neutral directory with filters
- **Messages:** Secure chat with lawyers
- **Documents:** Case files and shared documents
- **AI Assistant:** Legal Q&A with disclaimers
- **Court Calendar:** Upcoming hearings and deadlines
- **Legal Research:** NJDG judgment search

Key Metrics:

- Active cases count
- Pending actions

- Upcoming hearings
- Message notifications

Lawyer Dashboard

Main Sections:

- **Case Pipeline:** New inquiries, active matters
- **Client Communications:** Secure messaging hub
- **Calendar Management:** Court dates, client meetings
- **Document Center:** Case files, templates, forms
- **Research Tools:** NJDG search, case analytics
- **Practice Analytics:** Time tracking, earnings (optional)
- **Compliance Center:** DPDP requests, audit logs

Key Metrics:

- Active cases
- Response time to clients
- Upcoming deadlines
- Revenue tracking (optional)

Admin Dashboard

Main Sections:

- **User Management:** KYC approvals, account status
- **Compliance Monitor:** BCI violations, DPDP requests
- **Platform Analytics:** User engagement, feature usage
- **Content Moderation:** Profile reviews, flag handling
- **System Health:** API status, performance metrics
- **Audit Trails:** All platform activities log

Key Metrics:

- User growth (clients/lawyers)
- Platform usage statistics
- Compliance incident tracking
- System uptime and performance

Backend Architecture

Microservices Structure

```
|— Auth Service (User management, JWT, MFA)
|— Profile Service (KYC, verification, compliance)
|— Matching Service (Lawyer discovery, filtering)
|— Communication Service (Messaging, calls, encryption)
|— Case Service (Matter management, workflows)
|— Document Service (File storage, version control)
|— Search Service (NJDG integration, judgment search)
|— AI Service (Legal assistant, NLP processing)
|— Compliance Service (DPDP, consent, audit)
|— Analytics Service (Usage tracking, insights)
|— Notification Service (Email, SMS, push)
|— Payment Service (Billing, transactions)
```

Database Design

Primary Database: PostgreSQL with read replicas

```
-- Core user tables
users (id, email, phone, role, created_at, updated_at)
user_profiles (user_id, name, address, kyc_status, verification_docs)
lawyer_profiles (user_id, bar_enrollment, practice_areas, experience_years)

-- Case management
cases (id, client_id, lawyer_id, type, status, court, created_at)
case_documents (id, case_id, filename, file_path, uploaded_by, timestamp)
case_activities (id, case_id, activity_type, description, timestamp)

-- Communication
conversations (id, case_id, participants, encryption_key)
messages (id, conversation_id, sender_id, content_encrypted, timestamp)

-- NJDG integration
judgments (id, court, case_number, parties, date, citation, file_url)
judgment_metadata (judgment_id, keywords, practice_areas, indexed_content)

-- Compliance
consent_records (user_id, purpose, granted_at, withdrawn_at, policy_version)
audit_logs (id, user_id, action, resource, timestamp, ip_address)
```

Additional Storage:

- **Redis:** Caching, session management, message queues
- **Elasticsearch:** Judgment search, full-text indexing
- **AWS S3/Azure Blob:** Document storage, file versioning
- **MinIO:** On-premise object storage for sensitive documents

API Architecture

RESTful APIs with GraphQL for complex queries

```
/api/v1/auth/* - Authentication endpoints  
/api/v1/users/* - User management  
/api/v1/lawyers/* - Lawyer operations  
/api/v1/cases/* - Case management  
/api/v1/messages/* - Communication  
/api/v1/search/* - NJDG and judgment search  
/api/v1/ai/* - Legal assistant  
/api/v1/compliance/* - DPDP operations
```

Step-by-Step Implementation Plan

Weeks 1-2: Project Setup & Infrastructure

Deliverables:

- Development environment setup (Docker, Kubernetes)
- CI/CD pipeline (GitHub Actions/GitLab CI)
- Cloud infrastructure (AWS/Azure/GCP)
- Database design and initial schemas
- Basic authentication service

Team Tasks:

- DevOps: Infrastructure setup, monitoring
- Backend: Auth service, database setup
- Frontend: Project structure, UI component library
- Compliance: DPDP policy drafting

Weeks 3-4: User Management & Profiles

Deliverables:

- Multi-role registration and login
- KYC document upload and verification
- Basic profile management
- Email/SMS verification
- Admin panel for user approval

Technical Focus:

- JWT token management
- File upload handling

- OCR integration for documents
- Role-based access control

Weeks 5-6: Lawyer Directory & Matching

Deliverables:

- Neutral lawyer discovery interface
- Practice area filtering
- Location-based search
- Lawyer profile verification workflow
- BCI compliance monitoring

Key Considerations:

- No promotional rankings or paid placements
- Verification badges only
- Compliance flagging system

Weeks 7-8: Communication & Basic Case Management

Deliverables:

- End-to-end encrypted messaging
- File sharing capabilities
- Basic case intake forms
- Case status tracking
- Document upload and management

Security Focus:

- Signal Protocol implementation
- Message encryption at rest
- Audit logging for all communications

Weeks 9-12: NJDG Integration & AI Assistant

Deliverables:

- NJDG judgment search interface
- Metadata extraction and indexing
- AI legal assistant with disclaimers
- Confidence scoring system
- Citation tracking

Technical Challenges:

- NJDG API rate limiting
- Large language model integration
- Vector similarity search
- Response quality assurance

Weeks 13-16: Enhanced Features & Mobile

Deliverables:

- Advanced case workflows
- Court hearing management
- Mobile web optimization
- Payment processing integration
- Analytics dashboard

Weeks 17-20: Advanced Analytics & Mobile Apps

Deliverables:

- Predictive case analytics
- Native mobile applications
- Enterprise features
- API documentation
- Third-party integrations

Weeks 21-24: Testing, Security & Launch

Deliverables:

- Comprehensive security testing
- Penetration testing
- Performance optimization
- User acceptance testing
- Soft launch with beta users

Compliance Framework

BCI Rule 36 Compliance

Restrictions:

- No lawyer advertising or promotional content
- No paid rankings or featured listings
- No direct solicitation features
- No success rate or win/loss statistics

Implementation:

- Content moderation system
- Automated flag detection
- Manual review workflows
- Compliance training for lawyers

DPDP Act 2023 Compliance

Requirements:

- Explicit consent for data collection
- Data localization within India
- Data subject rights portal
- Breach notification procedures
- Data Protection Officer appointment

Technical Implementation:

- Granular consent management
- Automated DSR fulfillment
- Incident response procedures
- Regular compliance audits

Technology Stack

Backend

- **Framework:** Node.js with Express/NestJS or Python with FastAPI
- **Database:** PostgreSQL with Redis caching
- **Search:** Elasticsearch/OpenSearch
- **Queue:** Redis/RabbitMQ
- **Storage:** AWS S3/Azure Blob Storage

Frontend

- **Web:** React with TypeScript or Next.js
- **Mobile:** React Native or Flutter
- **UI Library:** Material-UI or Ant Design
- **State Management:** Redux Toolkit/Zustand

Infrastructure

- **Cloud:** AWS/Azure/GCP with Indian data centers
- **Containers:** Docker with Kubernetes orchestration
- **Monitoring:** Prometheus, Grafana, ELK Stack
- **Security:** WAF, DDoS protection, SSL/TLS

Agent Prompt Pack for Judicature Features

```
# Judicature MVP - Feature Development Prompts
```

```
## Global Guidelines
```

```
**CRITICAL COMPLIANCE RULES:**
```

- BCI Rule 36: NO advertising/promotional content for lawyers
- DPDP Act: Explicit consent required for all data collection
- Security: All communications must be end-to-end encrypted
- Disclaimers: AI responses must clearly state "This is not legal advice"

```
---
```

```
## 1. Authentication System
```

```
### Prompt:
```

Build a secure multi-role authentication system for a legal tech platform with the following features:

```
**Features:**
```

- Support for Client, Lawyer, and Admin roles
- Email and phone number registration
- Multi-factor authentication (OTP via SMS/Email)
- JWT token management with refresh tokens
- Password reset functionality
- Session management and logout

```
**Compliance Requirements:**
```

- DPDP Act compliant consent capture during registration
- Audit logging for all authentication events
- Rate limiting for login attempts
- Strong password policies

```
**Technical Specs:**
```

- Node.js/Express or Python/FastAPI backend
- PostgreSQL database
- Redis for session management

- JWT tokens with 15-minute access + 7-day refresh
- bcrypt for password hashing

****Database Schema:****

users (id, email, phone, password_hash, role, email_verified, phone_verified, created_at)

user_sessions (id, user_id, refresh_token, expires_at, device_info)

auth_logs (id, user_id, action, ip_address, user_agent, timestamp)

****Deliverables:****

- REST API endpoints for auth operations
- Input validation and sanitization
- Error handling with appropriate status codes
- API documentation with examples

2. Lawyer Profile Management

Prompt:

Create a lawyer profile system that complies with Bar Council of India regulations while

****BCI Rule 36 Compliance:****

- NO promotional language or claims of expertise
- NO advertising or solicitation features
- NO rankings or "top lawyer" listings
- Only factual information: education, experience years, practice areas

****Features:****

- Bar Council enrollment verification
- Document upload for credentials
- Practice area selection (from predefined list)
- Court jurisdiction selection
- Experience years (number only, no claims)
- Contact information and availability status

****Verification Workflow:****

- Manual review of all lawyer registrations
- Bar Council enrollment number validation
- Document verification by admin team
- Approval/rejection with feedback

****Technical Requirements:****

- File upload with virus scanning
- OCR processing for document verification
- Automated compliance checking for promotional content
- Profile status tracking (pending/verified/rejected)

****Database Schema:****

lawyer_profiles (

user_id, bar_enrollment_number, bar_state,

```
experience_years, education, practice_areas[],
court_jurisdictions[], verification_status,
verification_documents[], verified_at, verified_by
)
```

****Content Moderation:****

- Keyword filtering for promotional terms
- Manual review queue for flagged content
- Automatic rejection of advertising language

3. Secure Communication System

Prompt:

Build an end-to-end encrypted messaging system for attorney-client communications with fu

****Security Requirements:****

- End-to-end encryption using Signal Protocol or similar
- Perfect forward secrecy for all messages
- No server-side message storage in plaintext
- Secure file sharing with encryption
- Message deletion and retention policies

****Features:****

- Real-time messaging with WebSocket
- File attachment support (documents, images)
- Message status indicators (sent, delivered, read)
- Conversation threads linked to cases
- Message search within conversations
- Export functionality with client consent

****Technical Implementation:****

- WebSocket connection for real-time communication
- Message encryption before database storage
- Key exchange and rotation mechanisms
- Secure file upload and sharing
- Message indexing for search (encrypted metadata only)

****Database Schema:****

```
conversations (id, case_id, participants[], encryption_key_id, created_at)
messages (id, conversation_id, sender_id, encrypted_content, metadata, timestamp)
message_keys (id, conversation_id, key_data, created_at, rotated_at)
```

****Client-Side Requirements:****

- Message encryption/decryption in browser
- Key management and storage
- Offline message queue
- File encryption before upload

4. Case Management System

Prompt:

Develop a comprehensive case management system for legal professionals with client access

Core Features:

- Case creation with intake form wizard
- Matter lifecycle tracking (stages: consultation, active, closed)
- Task management with deadlines and reminders
- Document repository with version control
- Court hearing calendar integration
- Client status updates and notifications
- Time tracking (optional for billing)

User Roles:

- Lawyers: Full case management access
- Clients: View-only access to their cases
- Staff: Limited access based on assignments

Workflow Management:

- Customizable case stages and milestones
- Automated status updates based on actions
- Deadline tracking with escalation
- Template documents for common case types

Technical Requirements:

- RESTful API for case operations
- File storage with access control
- Calendar integration (Google Calendar, Outlook)
- Email notifications for updates
- Search functionality across cases and documents

Database Schema:

cases (id, client_id, lawyer_id, case_number, type, status, court, created_at)

case_stages (id, case_id, stage_name, completed_at, notes)

case_documents (id, case_id, filename, file_path, uploaded_by, version, timestamp)

case_tasks (id, case_id, title, description, due_date, assigned_to, status)

case_hearings (id, case_id, court_name, hearing_date, hearing_type, notes)

5. NJDG Integration & Judgment Search

Prompt:

Create a comprehensive judgment search system integrated with India's National Judicial C

Features:

- Search across Supreme Court, High Courts, and District Courts
- Advanced filtering (court, date range, party names, citation)
- Full-text search within judgment content

- Bookmark and save functionality for users
- Citation analysis and cross-references
- Download and sharing capabilities
- Search history and analytics

****Technical Implementation:****

- NJDG API integration with proper authentication
- Elasticsearch/OpenSearch for indexing and search
- OCR processing for scanned judgment PDFs
- Metadata extraction and tagging
- Relevance scoring and ranking algorithms

****Search Capabilities:****

- Boolean search with AND/OR/NOT operators
- Phrase searching with quotes
- Wildcard and fuzzy search
- Filter by court level, jurisdiction, date ranges
- Sort by relevance, date, court hierarchy

****Database Schema:****

judgments (id, court_id, case_number, parties, judge_names, date, citation)
 judgment_content (judgment_id, full_text, summary, keywords[])
 judgment_metadata (judgment_id, practice_areas[], legal_topics[], precedent_value)
 user_bookmarks (user_id, judgment_id, notes, created_at)

****API Endpoints:****

- `/api/search/judgments` - Main search endpoint
- `/api/judgments/:id` - Get specific judgment
- `/api/bookmarks` - User bookmark management
- `/api/search/suggestions` - Search suggestions and autocomplete

6. AI Legal Assistant

Prompt:
 Build an AI-powered legal assistant that provides preliminary legal guidance while maintaining accuracy and ethical standards.

****CRITICAL DISCLAIMERS:****

- Every response must include: "This is preliminary guidance only and does not constitute legal advice."
- Must recommend consulting with a qualified lawyer for specific legal matters
- Cannot provide definitive legal opinions or case outcomes
- Must cite sources for all legal information provided

****Features:****

- Natural language processing for legal queries
- Multi-language support (English, Hindi, regional languages)
- Case law similarity matching
- Confidence scoring for response reliability
- Integration with judgment database for citations
- Query categorization and routing
- Escalation to human lawyers for complex queries


```

**Technical Architecture:**
- Large Language Model (GPT-4, Claude, or fine-tuned model)
- RAG (Retrieval Augmented Generation) with legal document corpus
- Vector database for judgment embeddings
- Confidence threshold system
- Response validation and filtering

**Response Format:**
- Clear disclaimer at the beginning
- Main response with confidence score
- Relevant case law citations
- Recommendation to consult lawyer
- Option to connect with verified lawyers

**Implementation:**

```

```

class LegalAssistant:
def process_query(self, query, user_context):
# 1. Query validation and sanitization
# 2. Intent classification
# 3. Retrieve relevant case law
# 4. Generate response with LLM
# 5. Confidence scoring
# 6. Add mandatory disclaimers
# 7. Citation formatting
return formatted_response

```

```

**Database Schema:**

```

```

ai_conversations (id, user_id, session_id, created_at)
ai_queries (id, conversation_id, user_query, ai_response, confidence_score, timestamp)
query_citations (query_id, judgment_id, relevance_score)

```

```

---

```

```

### 7. DPDP Compliance Module

```

```

### Prompt:

```

```

Implement a comprehensive data protection system compliant with India's Digital Personal

```

```

**Core Requirements:**
- Explicit and informed consent collection
- Granular purpose-based consent management
- Data subject rights (DSR) portal
- Consent withdrawal mechanisms
- Data breach notification system
- Cross-border data transfer controls

```

```

**Consent Management:**
- Purpose-based consent categories
- Clear, plain language explanations
- Opt-in only (no pre-checked boxes)
- Consent versioning and audit trail
- Easy withdrawal process
- Parental consent for minors

**Data Subject Rights:**
- Right to access personal data
- Right to correction/rectification
- Right to data portability
- Right to erasure/deletion
- Right to grievance redressal

**Technical Implementation:**
- Consent capture with timestamp and IP logging
- Automated DSR fulfillment workflows
- Data inventory and mapping
- Encryption for all personal data
- Access controls and audit logging

**Database Schema:**

```

consent_records (user_id, purpose_category, consent_given, timestamp, ip_address, policy_version)

dsr_requests (id, user_id, request_type, status, requested_at, fulfilled_at, response_data)

data_processing_log (id, user_id, processing_purpose, data_types[], timestamp, legal_basis)

```

**API Endpoints:**
- `/api/consent/grant` - Record user consent
- `/api/consent/withdraw` - Process consent withdrawal
- `/api/dsr/request` - Submit data subject request
- `/api/dsr/status/:id` - Check request status
- `/api/privacy/export` - Export user data

```

8. Payment Processing System

Prompt:

Create a secure payment system for legal consultations and services with Indian compliance

```

**Compliance Requirements:**
- RBI guidelines for payment aggregators
- PCI DSS compliance for card data
- Indian tax regulations (GST)
- Anti-money laundering (AML) checks
- Know Your Customer (KYC) verification

```

```

**Features:**
- Multiple payment methods (cards, UPI, net banking, wallets)
- Subscription management for lawyers

```

- Transaction fees and commission calculation
- Refund and dispute management
- Invoice generation with GST
- Payout management for lawyers

****Technical Integration:****

- Payment gateway integration (Razorpay, Stripe, PayU)
- Webhook handling for payment status
- Encrypted storage of payment metadata
- Audit trail for all financial transactions

****Security Measures:****

- No storage of card details (tokenization only)
- Secure API endpoints with rate limiting
- Transaction monitoring for fraud detection
- Regular security audits

****Database Schema:****

transactions (id, user_id, amount, currency, status, payment_method, gateway_transaction_id)
 subscriptions (id, lawyer_id, plan_type, amount, billing_cycle, status, next_billing_date)
 payouts (id, lawyer_id, amount, status, processed_at, bank_details_id)
 invoices (id, transaction_id, invoice_number, gst_amount, download_url)

9. Analytics Dashboard

Prompt:

Build comprehensive analytics dashboards for different user roles with privacy-compliant

****Client Analytics:****

- Case progress tracking
- Communication statistics
- Document access logs
- AI assistant usage
- Lawyer interaction metrics

****Lawyer Analytics:****

- Case load and pipeline
- Client response times
- Revenue tracking (optional)
- Time utilization reports
- Client satisfaction scores

****Platform Analytics (Admin):****

- User acquisition and retention
- Feature adoption rates
- Platform performance metrics
- Compliance incident tracking
- Revenue and financial metrics

****Technical Requirements:****

- Real-time data processing with Apache Kafka/Redis Streams
- Time-series database (InfluxDB/TimescaleDB) for metrics
- Dashboard framework (Grafana, custom React dashboards)
- Data anonymization for privacy compliance
- Export functionality for reports

****Privacy Considerations:****

- Aggregate data only (no individual user tracking)
- Consent-based analytics collection
- Data retention policies
- Opt-out mechanisms for users

****Implementation:****

```
// Analytics event tracking
const trackEvent = (eventType, userId, metadata) => {
// Anonymize sensitive data
// Validate consent preferences
// Store in time-series database
// Trigger real-time dashboard updates
}
```

10. Mobile Application

Prompt:

Develop native mobile applications (iOS/Android) for Judicature with all core platform fe

****Core Features:****

- Secure login with biometric authentication
- Push notifications for case updates
- Mobile-optimized messaging interface
- Document camera for file uploads
- Offline mode for viewing saved content
- Location-based lawyer discovery
- Voice-to-text for AI assistant queries

****Technical Stack Options:****

- React Native with TypeScript
- Flutter with Dart
- Native development (Swift/iOS, Kotlin/Android)

****Security Requirements:****

- Certificate pinning for API calls
- Local data encryption using device keystore
- Secure storage for authentication tokens
- App integrity verification
- Remote wipe capabilities for lost devices

****Mobile-Specific Features:****

- Fingerprint/Face ID authentication
- Camera integration for document scanning

- GPS location for nearby lawyers
- Push notification handling
- Background sync for messages
- App shortcuts and widgets

****API Considerations:****

- RESTful APIs with mobile-optimized responses
- GraphQL for efficient data fetching
- Offline-first architecture with sync
- Image compression and optimization
- Pagination for large data sets

Development Guidelines

Code Quality Standards:

- TypeScript/ESLint configuration
- Unit test coverage >80%
- Integration tests for critical paths
- Security vulnerability scanning
- Performance monitoring and optimization

Documentation Requirements:

- API documentation with examples
- Database schema documentation
- Deployment and configuration guides
- Security architecture documentation
- Compliance audit trail documentation

Testing Strategy:

- Unit tests for business logic
- Integration tests for APIs
- End-to-end tests for user workflows
- Security penetration testing
- Performance and load testing

Deployment Pipeline:

- CI/CD with automated testing
- Staging environment for user testing
- Blue-green deployment for zero downtime
- Database migration scripts
- Infrastructure as code (Terraform/CloudFormation)

This comprehensive MVP plan provides you with a complete roadmap for building Judicature, including detailed technical specifications, compliance requirements, and ready-to-use prompts for development teams. The phased approach ensures you can launch quickly while building towards a comprehensive legal technology platform that serves all stakeholders in the Indian legal ecosystem.



