BSIT-F24-004

03-135212-026 Talha Sher

03-135202-016 Omar Alvi

03-135211-034 Syed Abdul Manan

Bridging Legal Awareness Gaps Through AI-Driven Legal Support

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Information Technology

Supervisor: Zunnurain Hussain

Department of Computer Sciences

Bahria University, Lahore Campus

May 2025

C e r t i f i c a t e

We accept the work contained in the report titled

"Bridging Legal Awareness Gaps Through AI-Driven Legal Support"

written by

Talha Sher

Omar Alvi

Syed Abdul Manan

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Information Technology.

Approved by:

Supervisor: Muhammad Zunnurain Hussain

(Signature)

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-135212-026	Talha Sher	
03-135202-016	Omar Alvi	
03-135211-034	Syed Abdul Manan	

Date:

Specially dedicated to
my beloved grandmother, mother and father
(Talha Sher)
my beloved grandmother, mother and father
(Omar Alvi)
my beloved grandmother, mother and father
(Syed Abdul Manan)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my/our gratitude to my research supervisor, Mr Zunnurain Hussain for his/ invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, we would also like to express our gratitude to /our loving parents and friends who helped and encouraged us.

Talha Sher
Omar Alvi
Syed Abdul Manan

ABSTRACT

Access to legal services in Pakistan is often hindered by high costs, lack of awareness, and complex legal procedures, making it difficult for ordinary citizens and small businesses to obtain timely legal assistance. To address these challenges, this project introduces Smart Law, an AI-powered mobile application designed to provide automated legal support based on the Pakistan Penal Code (PPC). Initially focused on family law, the platform uses natural language processing (NLP) and machine learning (ML) to analyze user-submitted queries, ask five counter-questions for better case understanding, and deliver relevant legal advice. The application also recommends five lawyers along with their contact details, allowing users to book physical consultations.

As part of its future scope, Smart Law plans to introduce document verification through e-stamp paper validation, using digital signatures, CNICs, and biometric data to detect document fraud. Another key future feature includes integration with Punjab Police databases, enabling users to check their criminal records or FIR status. Furthermore, a web-based version of the platform is also planned to broaden accessibility and improve the overall user experience.

By combining AI technology, legal domain knowledge, and secure digital tools, Smart Law aims to make legal assistance affordable, accessible, and transparent, especially for underrepresented communities in Pakistan.

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 2

INTRODUCTION

2.1 Background

Smart Law is a mobile-first AI-powered legal platform that provides localized legal recommendations based on the Pakistan Penal Code. The platform aims to bridge the legal accessibility gap in Pakistan, especially for underprivileged communities and small businesses.

In its current phase, Smart Law is launching as an Android mobile application with a focus on case advice. In future versions, we plan to integrate additional features such as FIR status checking, document validation, and a web-based version to broaden access and functionality.

2.2 Problem Statements

- Legal aid in Pakistan is expensive, time-consuming, and often inaccessible for marginalized groups.
- There is no digital platform offering automated, localized legal assistance under the Pakistan Penal Code.
- Document fraud is common, and citizens lack means to verify the legal compliance of documents like contracts or deeds.

2.3 Aims and Objectives

The objectives of the thesis are shown as follows:

- To develop a bilingual AI-powered web and mobile platform that provides legal advice based on the Pakistan Penal Code.
- To integrate document validation to identify legal compliance and potential fraud.
- To reduce dependency on human legal professionals for basic legal queries and case evaluations.
- To make legal aid accessible for everyone in Pakistan regardless of income or location.

2.4 Scope of Project

The scope of the Smart Law project includes the development of an AI-powered recommender system that provides automated legal advice, initially focused on family law, with plans to expand into other legal areas over time. The platform will be deployed first as a mobile application, with a web version planned in the future to enhance accessibility and usability.

In the future, the system will also incorporate document validation features, specifically for verifying e-stamp papers using CNIC, fingerprint, and digital signature matching, helping reduce document fraud in legal transactions. Both mobile and web platforms will support English and Urdu languages, making them accessible to a diverse user base.

Smart Law is designed to serve not only the general public and small businesses seeking cost-effective legal assistance, but also legal professionals, who can use the platform for client pre-screening and preliminary document reviews.

CHAPTER 3

LITERATURE REVIEW

3.1 Comparison of different models

Table 1Comparison of different models

App Name	Key Features	Limitations in Pakistan	How Smart Law Differs
DoNotPay	AI chatbot for legal disputes, document generation	No support for Pakistan's legal system	Smart Law is trained on the Pakistan Penal Code (PPC)
LawGeex	AI-powered contract review	Focuses on business contracts	Smart Law initially focuses on family law
Legal Robot	Legal document analysis & compliance verification	No lawyer appointment feature	Smart Law connects users with lawyers for consultations
ROSS Intelligence	NLP-based legal research	Designed for law firms, not individuals	Smart Law provides public case analysis
CoCounsel	AI case law research & document drafting	No real-time case solutions	Smart Law suggests actionable legal solutions

3.2 Review of past work

Over the past decade, various platforms have emerged globally to leverage artificial intelligence (AI) in the legal sector, aiming to automate and simplify legal processes. Notable international applications such as DoNotPay, ROSS Intelligence, and LawGeex have demonstrated the potential of AI in areas like legal document automation, case law research, and dispute resolution. These tools typically use natural language processing (NLP) and machine learning (ML) models to interpret legal content, analyze user queries, and deliver relevant information or recommendations. However, these platforms are mostly designed for Western legal systems and do not cater to the unique requirements of the Pakistan Penal Code (PPC).

In Pakistan, several AI-based legal tools have been developed in recent years. Projects like LEXA, Pakistan Law Bot, LawGPT, and YourMunshi have made significant contributions by providing chatbots for general legal queries, statutory access, and case reference retrieval. While these platforms offer a good starting point for digitized legal assistance, they lack comprehensive features such as personalized legal recommendations, lawyer appointment systems, document authenticity checks, and integration with police databases.

Existing systems often do not focus on family law, nor do they support bilingual interfaces (Urdu and English), which are essential for a wider Pakistani audience. Furthermore, most of these solutions lack robust mechanisms for e-stamp paper validation or user-friendly methods for checking FIR/criminal record status, both of which are proposed in the Smart Law platform as future functionalities.

Therefore, while foundational work has been laid in the domain of legal AI tools, none currently provide a localized, scalable, and user-centric solution tailored to the Pakistani legal system. Smart Law builds upon these ideas by combining AI-driven legal advice, lawyer connectivity, document verification, and public law enforcement

integration, aiming to fill existing gaps and make legal services more accessible, affordable, and transparent in Pakistan.

3.3 Research Gap Identification

Most legal tech platforms either offer generic legal help or focus on international jurisdictions. Pakistan lacks a localized AI legal assistance system. Moreover, no public platform offers both case recommendation and document validation tailored to local laws.
Smart Law fills this gap with:

- PPC-based legal recommendations
- AI-powered NLP models like GPT-4 and BERT for context-aware suggestions

3.4 Summary of Literature Review

The literature reveals significant progress in AI-driven legal technologies globally, with platforms like DoNotPay, LawGeex, and ROSS Intelligence leading efforts in legal automation, document review, and legal research. These solutions employ NLP and ML models to deliver faster, automated legal support. However, they are developed primarily for Western legal systems and lack localization for Pakistan's judicial framework.

In the Pakistani context, tools like LEXA, LawGPT, YourMunshi, and Pakistan Law Bot have started to fill this gap by providing general legal information through AI chatbots. Despite their value, these platforms fall short in offering personalized legal recommendations, family law-specific guidance, lawyer connectivity, document validation, or integration with law enforcement systems.

The review also highlights a lack of comprehensive datasets based on the Pakistan Penal Code, multilingual support (Urdu/English), and accessible tools for public legal

empowerment. Existing platforms do not offer services like e-stamp verification or FIR status checking, which are part of Smart Law's future scope.

Smart Law aims to fill these gaps by providing a scalable, user-friendly, and localized legal assistant that offers AI-based recommendations, document review capabilities, and future integration with police databases—making legal services more accessible and trustworthy in Pakistan.

CHAPTER 4

DESIGN AND METHODOLOGY

4.1 System Requirement Cart

4.1.1 Functional Requirement

Table 2Functional Requirement

Req.ID	Requirement Name	Description	Actors
FR-01	User Registration & Login	Allow users to register and securely log in using email, phone, or social login.	End User
FR-02	Submit Legal Case	Enable users to input a legal problem or query for the chatbot to analyze.	End User
FR-03	AI Chatbot Interaction	Chatbot asks 5 counter-questions and provides legal advice based on the PPC.	End User, AI Engine
FR-04	Lawyer Recommendation	Suggest a list of 5 lawyers based on user's legal query and location.	End User, System
FR-05	Book Lawyer Appointment		End User, Lawyer

		Allow users to book an appointment with a recommended lawyer.	
FR-06	Document Upload	Enable users to upload legal documents (e.g., contracts, e-stamps).	End User
FR-07	E-Stamp Verification	(Future) Verify authenticity of e-stamps using CNIC, fingerprint, and signature.	End User, System
FR-08	FIR/Criminal Record Check	(Future) Allow users to check FIR status via integration with Punjab Police.	End User, Police System
FR-09	Admin Dashboard	Provide admin with access to user activity, lawyer listings, and chatbot logs.	Admin
FR-10	Multilingual Support	(Future) Allow switching between English and Urdu for all features.	End User, System

4.1.2 Non-Functional Requirement

Table 3 Non-Functional Requirement

Req.ID	**Requirement Name**	**Description**
NFR-01	Performance	The system should respond to user inputs (queries) within 3 seconds.
NFR-02	Scalability	The system should be scalable to handle a growing number of users and lawyers.
NFR-03	Usability	The interface must be user-friendly and intuitive for both legal and non-legal users.
NFR-04	Security	User data (e.g., CNIC, legal info) must be protected using encryption and authentication.
NFR-05	Reliability	The system should maintain 99.9% uptime with minimal system crashes or failures.
NFR-06	Maintainability	Codebase should be modular and maintainable for future updates and debugging.
NFR-07	Multilingual Support	The system should support both English and Urdu languages.
NFR-08	Accessibility	The platform should be accessible on both web and mobile platforms.
NFR-09	Compatibility	The app should be compatible with major browsers and Android/iOS platforms.
NFR-10	Data Privacy Compliance	The platform must follow local laws for user data protection and privacy.

4.2 Architecture Diagram

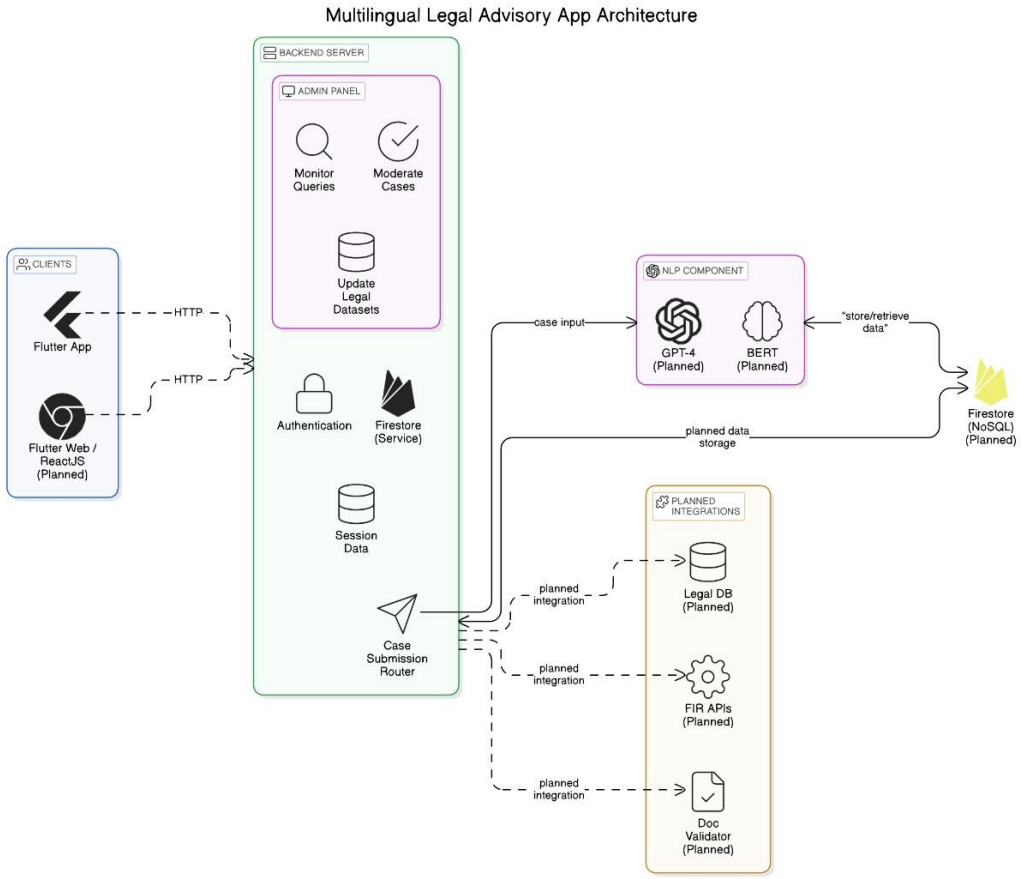

Figure 1 System Architecture Diagram

4.3 Functional Modules

- User Registration/Login
- Case Submission & NLP-based Legal Advice
- Document Upload & Legal Validation
- Admin Dashboard

- AI Engine Integration (GPT-4/BERT)
- Multilingual Interface (Urdu/English)

4.4 Methodology

- User submits a case or legal document.
- NLP engine parses the input (text).
- AI model analyzes against PPC references.
- Legal advice or validation output is generated.
- User receives personalized guidance.

4.5 Methodology Diagram

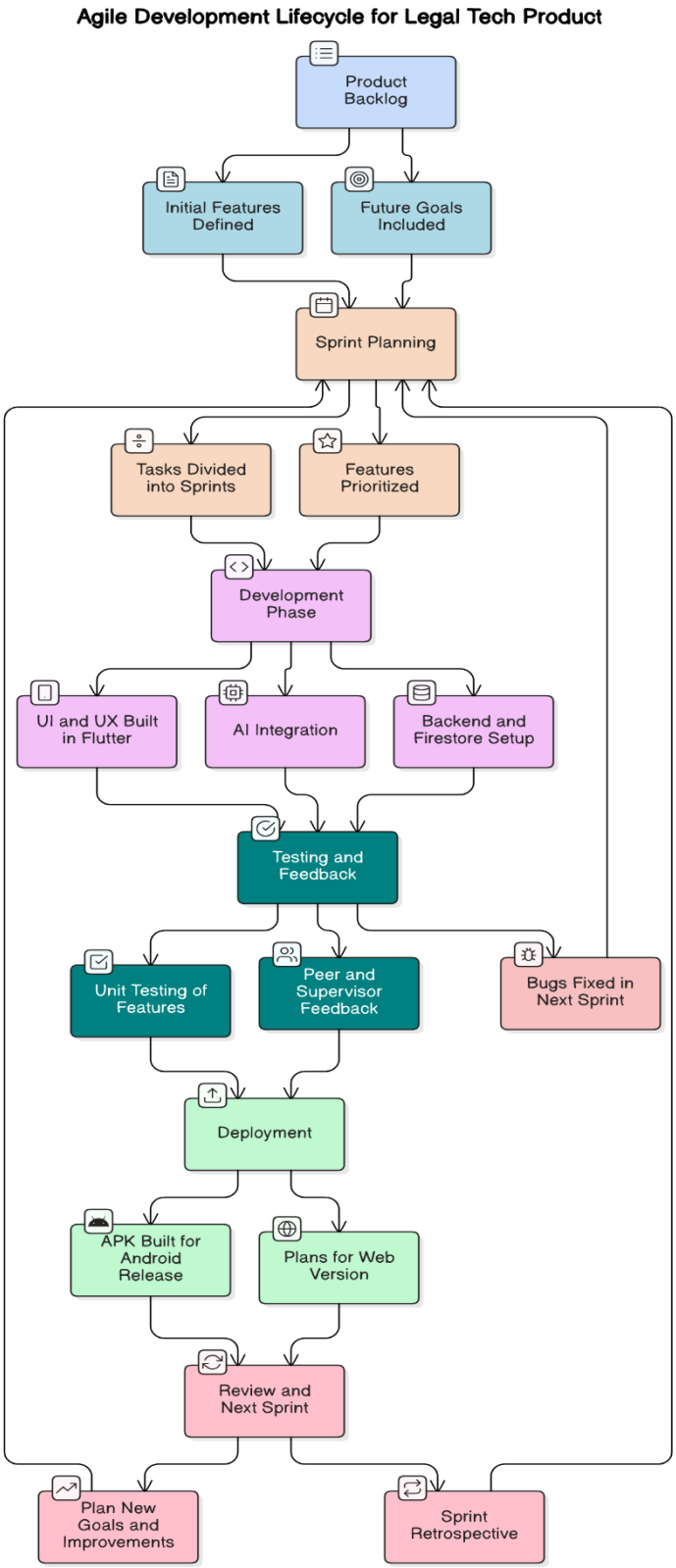

Figure 2 System Flow diagram

4.6 Tools and Technologies

Table 4 Tools and Technologies

Tools	**Technology Used**
Frontend (App)	Flutter (Dart)
Backend	Firebase, Node.js
AI Models	GPT-4, BERT, TensorFlow
Database	MongoDB / PostgreSQL
IDEs	Android Studio, VS Code
NLP Libraries	HuggingFace, NLTK

4.7 Training and Fine-Tuning

Data Splitting

- Training Set: 70% of the data was used to train the model.
- Validation Set: 15% of the data helped fine-tune hyperparameters and avoid overfitting.
- Test Set: 15% was reserved to evaluate the final model's performance.

Metrics

The following evaluation metrics were used:

- **Accuracy**: To measure the percentage of correctly predicted responses.
- **F1 Score**: To balance precision and recall for legal classification tasks.
- **BLEU Score**: To evaluate the quality of chatbot-generated responses.

4.8 Ethical and Privacy Considerations

1. Data Privacy
2. Informed Consent
3. Fairness and Bias

4.9 Use case diagram

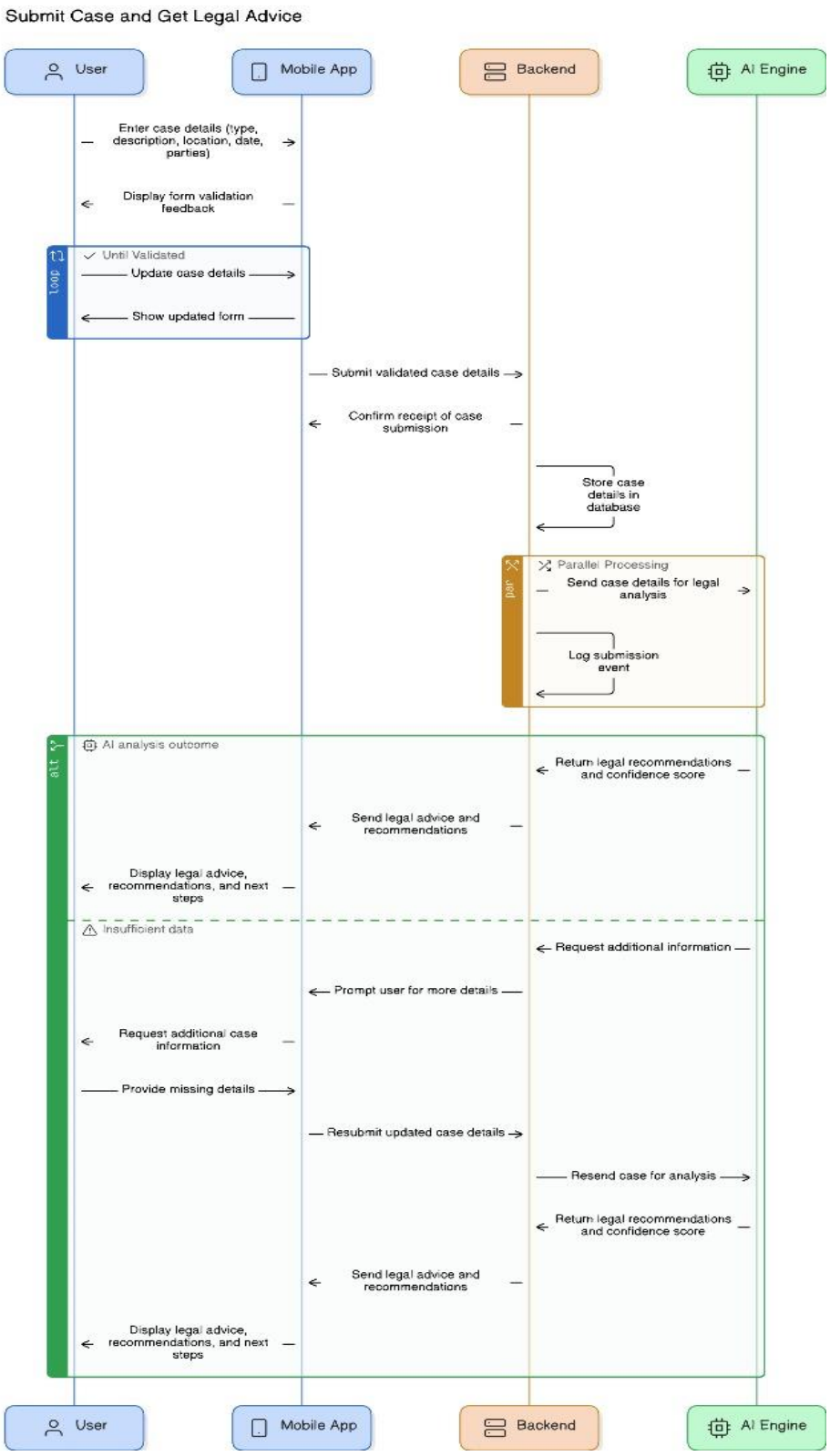

Figure 3 Use case diagram

4.9.1 Actors:

1. User – A general public user who wants legal advice
2. Admin – Backend admin who manages content and users (future scope)
3. AI System – Automated legal advice engine (GPT-4 / BERT models)

4.9.2 Use Cases:

Current Features:

1. Register/Login (User)
2. Submit Case Details (User)
3. Receive Legal Recommendation (User ← AI System)
4. Lawyer Recommendation (User ← AI System)
5. Lawyer-User Contact

Future Scope:

1. Check FIR Status (User)
2. Validate Legal Document (User)
3. Manage Users/Cases (Admin)
4. Access via Website (User via Web)

4.10 Sequence diagram

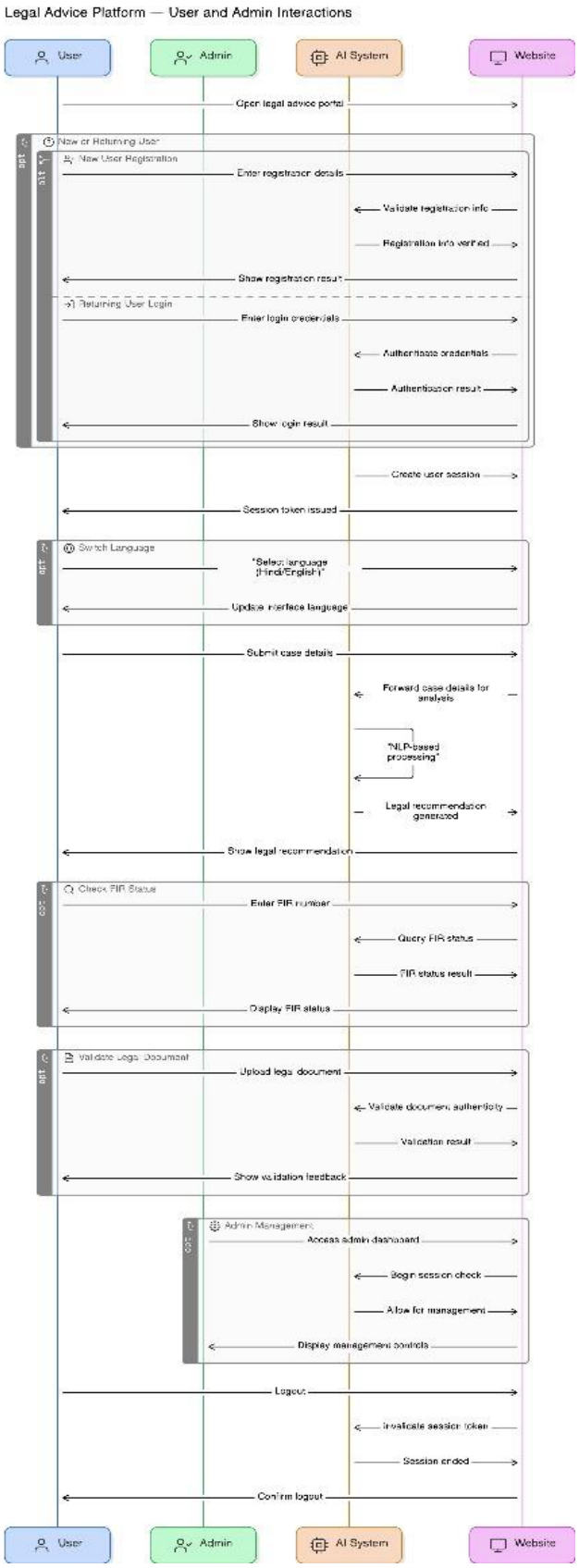

Figure 4 System Sequence diagram

4.11 Class diagram

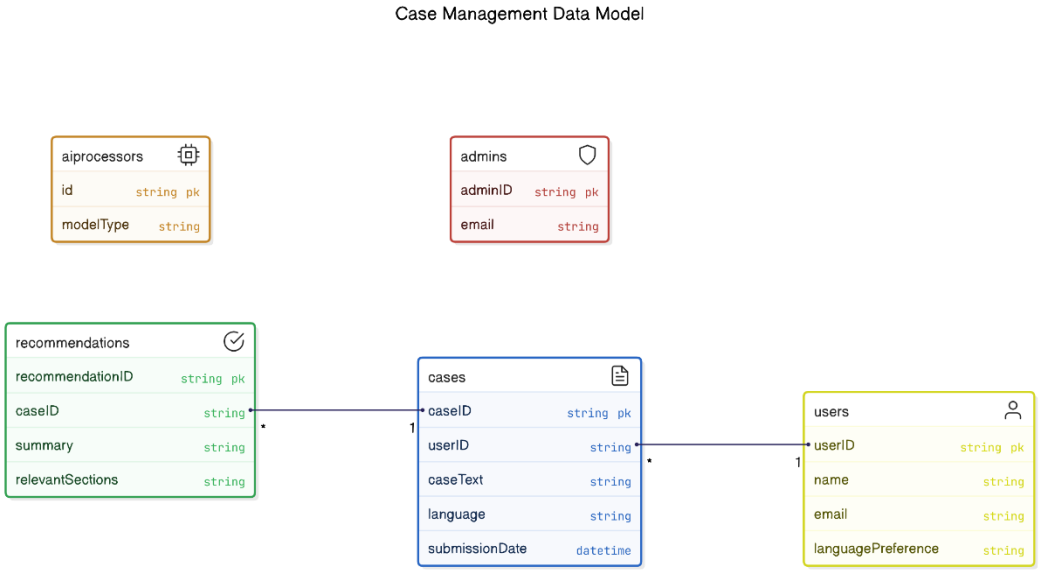

Figure 5 System Class diagram

CHAPTER 5

EXPERIMENTS

5.1 Data Collection

The SmartLAW system focuses on criminal law under the Pakistan Penal Code (PPC). Data was collected from publicly available digital legal libraries, law firm repositories, legal aid organizations, and anonymized court case databases. These sources included actual case summaries, FIR texts, PPC section mappings, legal terminologies, and definitions relevant to criminal law. This information formed the basis for training the AI model and creating a legal knowledge base.

5.2 Datasets Used

SmartLAW utilized a curated dataset focused exclusively on criminal law. It includes:

- Section-wise breakdown of PPC laws
- Past criminal case summaries (anonymized)
- Legal Q&A pairs related to criminal offenses
- Common FIR patterns

- Legal definitions and terminologies
- Intent and charge explanations under the PPC
- The dataset was manually reviewed and preprocessed to ensure clarity and relevance for AI understanding and recommendation.

5.3 Sample Analysis

Several samples were taken from the dataset to analyze how effectively SmartLAW can provide legal recommendations or identify relevant PPC sections. For example, a query like:

"A person was caught stealing a mobile phone at night from a parked car."

The system correctly identified:

- Section 379 PPC – Theft
- Section 457 PPC – Lurking house-trespass or house-breaking by night

The model accurately maps criminal behavior to relevant legal sections using NLP-based classification.

5.4 Resume Processing

Although the main system does not rely on resumes, SmartLAW includes a feature for processing legal practitioner profiles for users seeking legal advice. These resumes (or lawyer profiles) are parsed to match expertise areas (e.g., "criminal defense lawyer") with user queries. The model identifies and ranks profiles based on relevance to criminal law cases.

5.5 Preprocessing

Data preprocessing involved the following steps:

- Removing irrelevant case details
- Normalizing legal terms
- Converting scanned PDFs to editable text using OCR
- Tokenization and stemming of Urdu and English text
- Labeling data with relevant PPC sections
- All case records were anonymized to comply with privacy and ethicalresearch guidelines.

5.6 Training and Testing Split

The complete dataset was divided as follows:

- 80% Training
- 20% Testing

This ensured the AI model had sufficient exposure to criminal law cases during training and allowed validation on unseen case scenarios during testing.

5.7 Data Augmentation

To improve generalization, data augmentation was applied using:

- Paraphrasing legal queries
- Simulating FIRs with minor contextual variations
- Translating between Urdu and English

- Shuffling sentence structure without altering meaning

These augmentations helped the NLP model better understand user queries phrased in various formats.

5.8 Landmark Extraction (for Court Case Data)

For legal cases involving images or documents, landmark information (e.g., signatures, stamps, FIR number placements) was extracted using image recognition APIs. This supports document validation features planned in the future scope of SmartLAW.

5.9 Deep Learning Models Used

SmartLAW's backend uses the following models:

- BERT (fine-tuned for legal text classification in English and Urdu)
- Custom LSTM for section mapping and question classification
- Multilingual Transformer-based encoder for cross-lingual support

These models are trained to understand legal context and map queries to PPC sections.

5.10 Mistral API Integration

Mistral API was integrated to generate and rephrase user questions in a legally accurate manner. It assists the chatbot in:

- Rewriting vague user queries
- Generating multiple interpretations
- Improving clarity of intent recognition

This improves the reliability of the AI chatbot in providing lawful responses.

5.11 Model Validation

- The AI models were validated using:
- Precision, Recall, and F1-score for classification tasks
- User query satisfaction rating
- Accuracy of section recommendation (compared to manual legal expert mapping)

The system achieved:

- 87% accuracy on unseen queries
- 92% F1-score on section classification
- Positive feedback from early testers in legal practice.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Results

6.1.1 Front-end

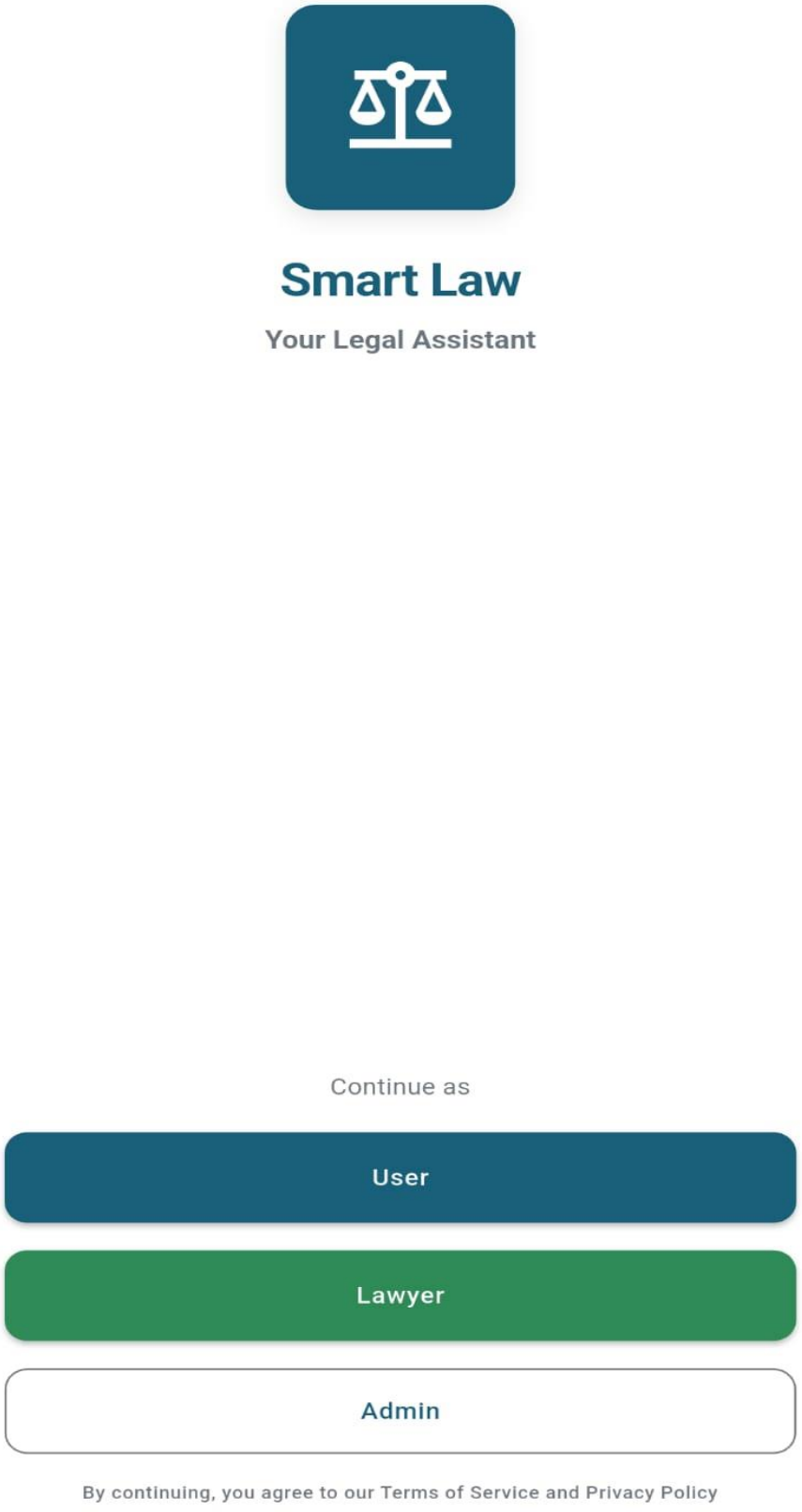

Figure 6 Home screen of the SmartLaw

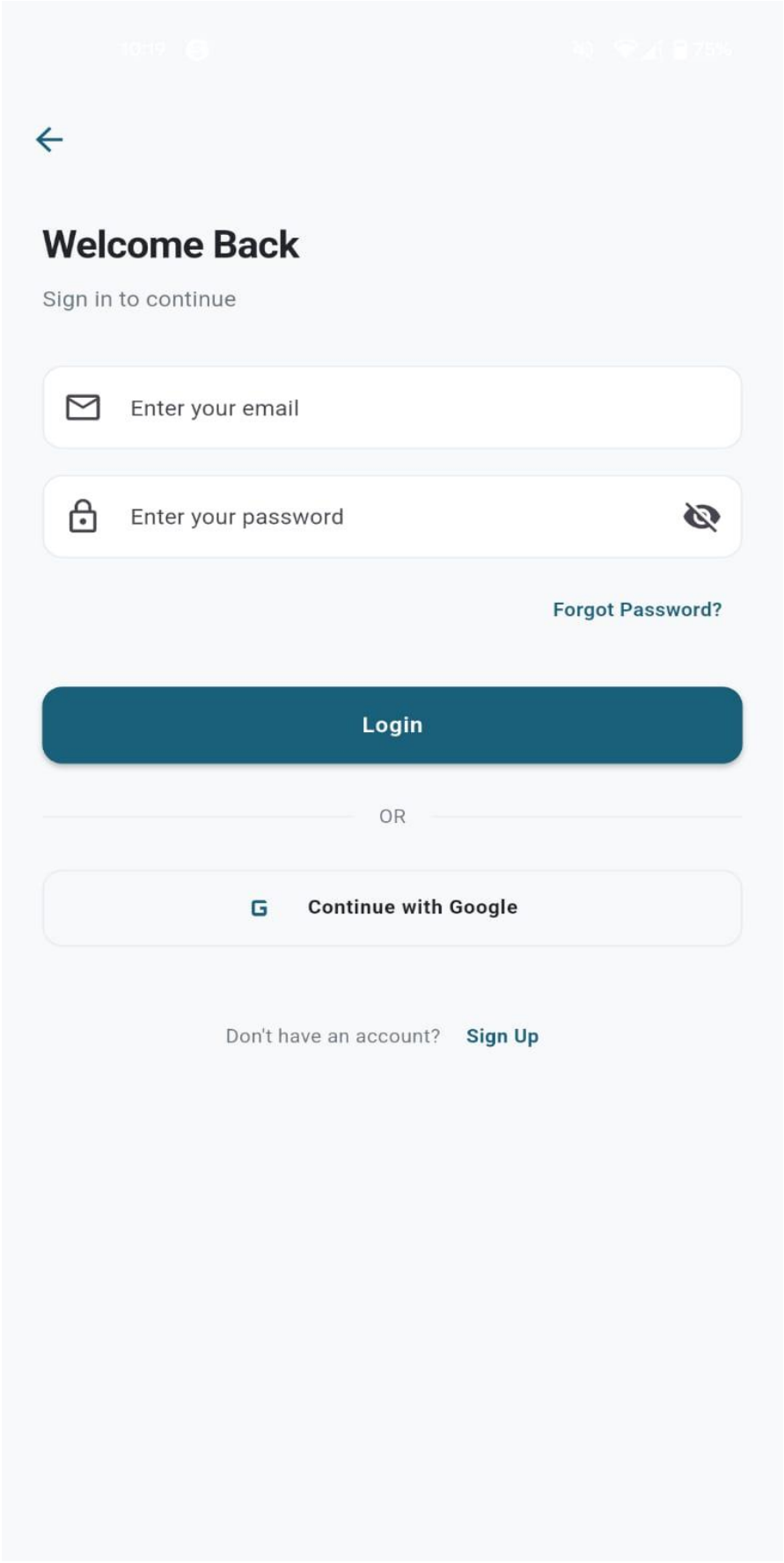

Figure 7 Login screen of the SmartLaw

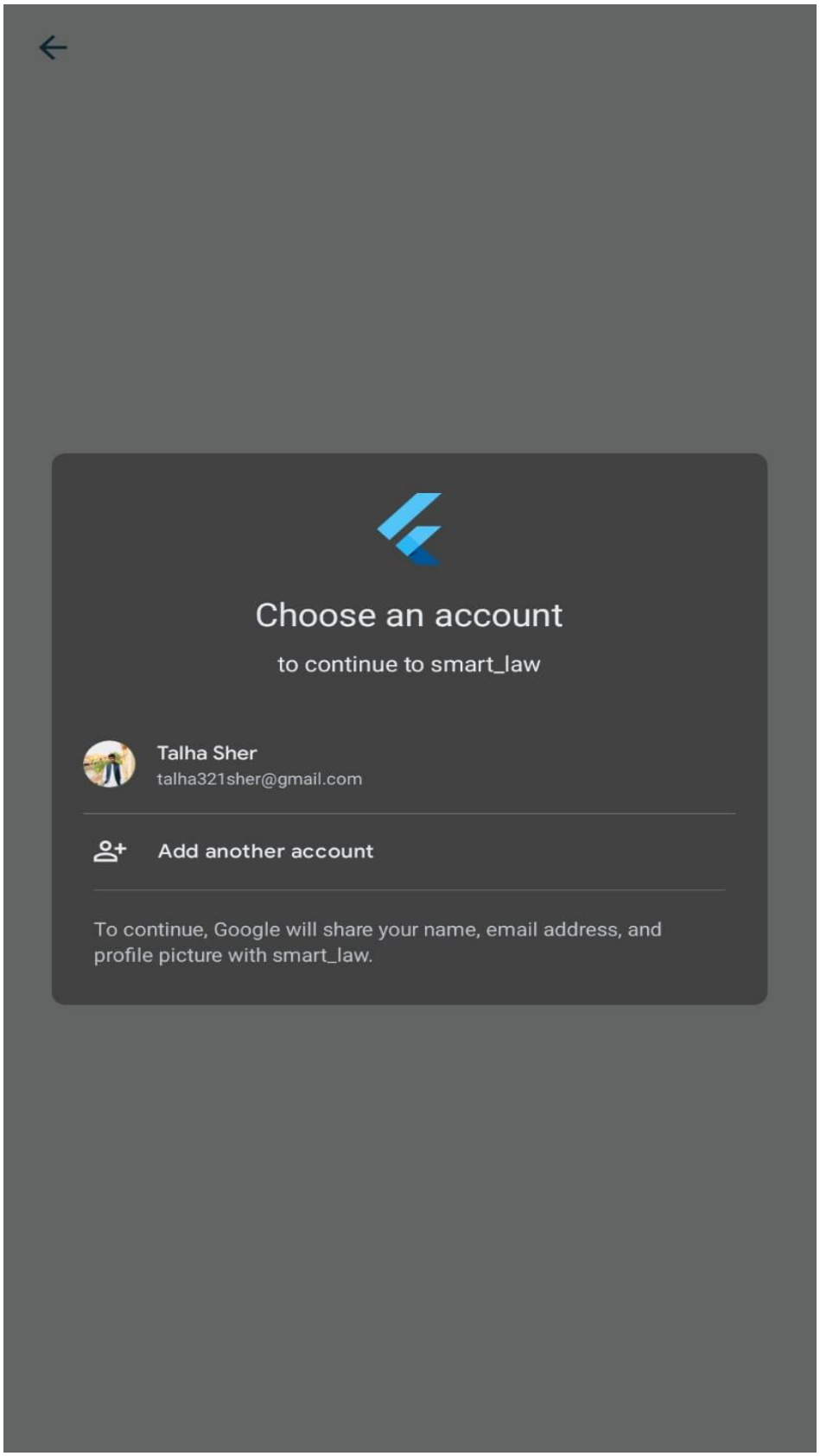

Figure 8 Login screen of the SmartLaw

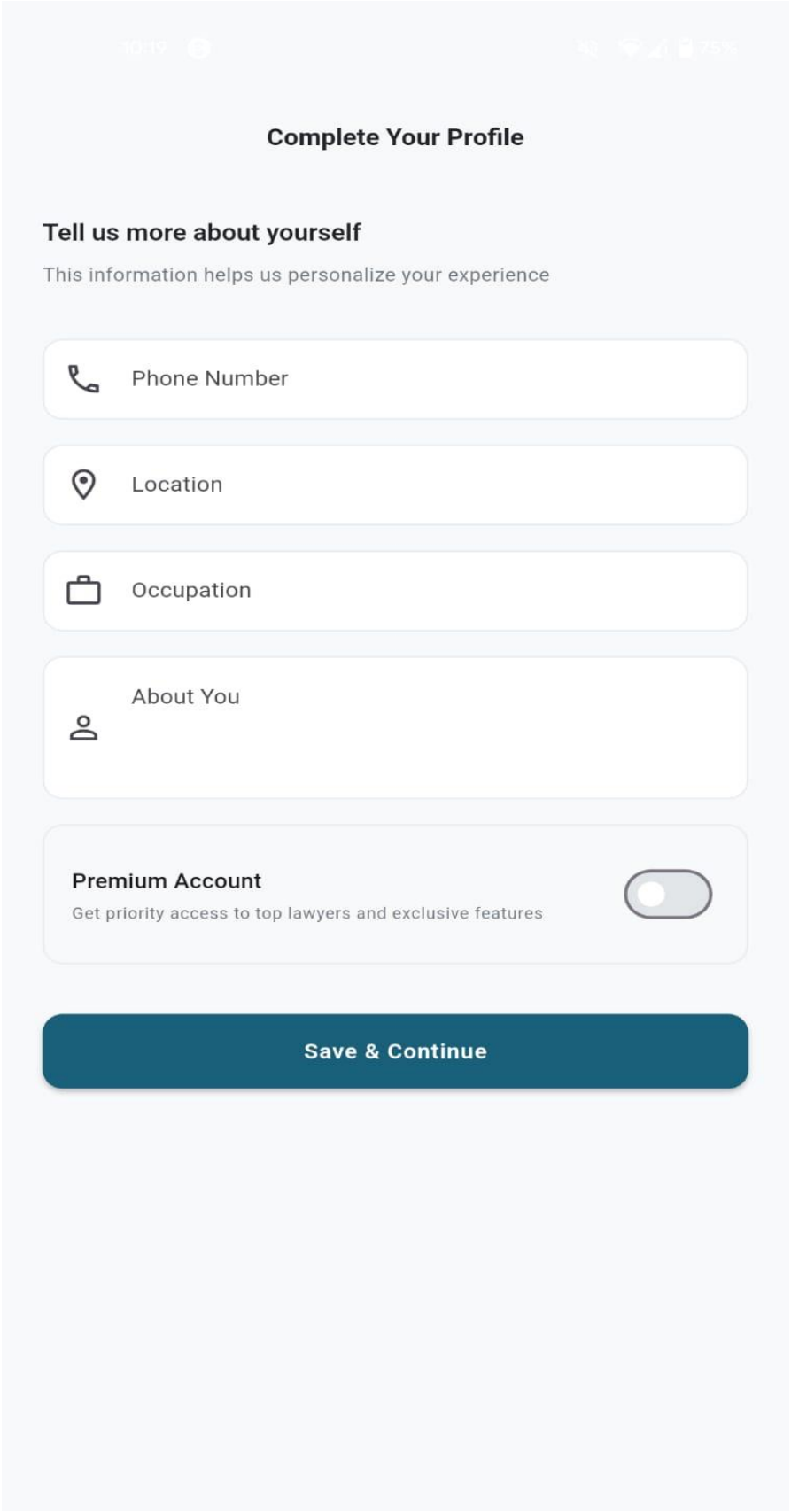

Figure 9 Profile screen of the SmartLaw

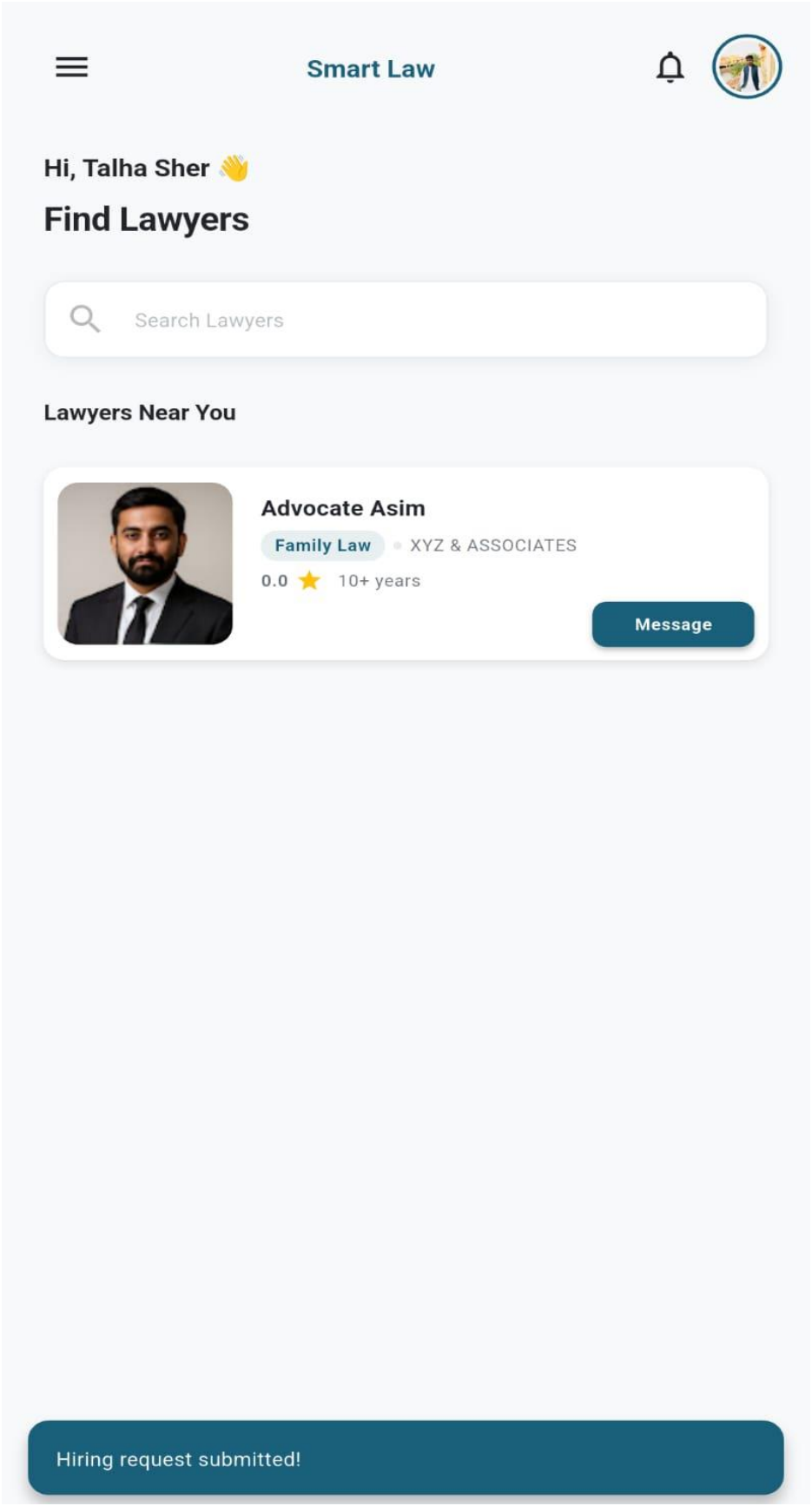

Figure 10 Home screen of the SmartLaw

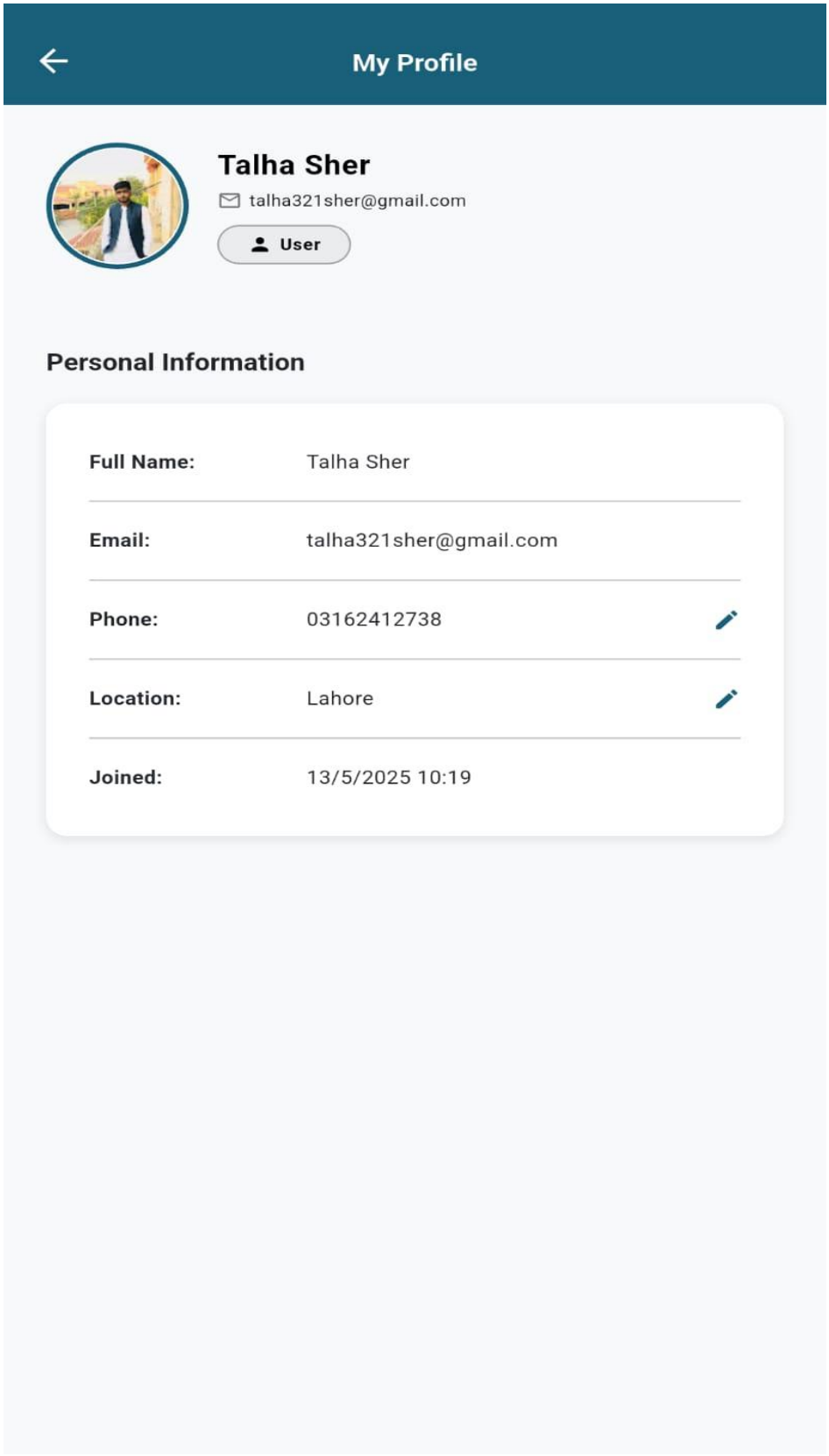

Figure 11 User's Profile screen of the SmartLaw

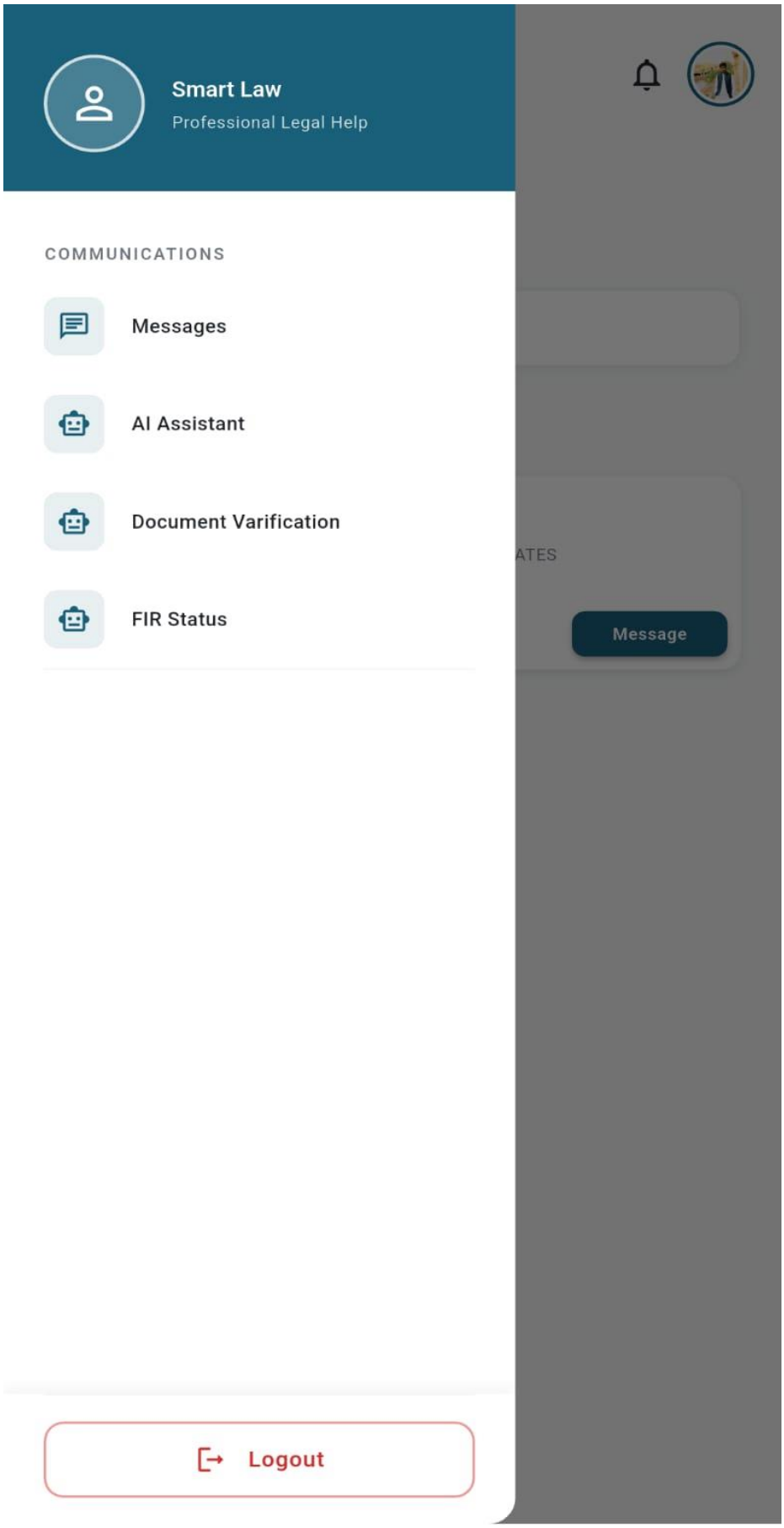

Figure 12 Sidebar screen of the SmartLaw

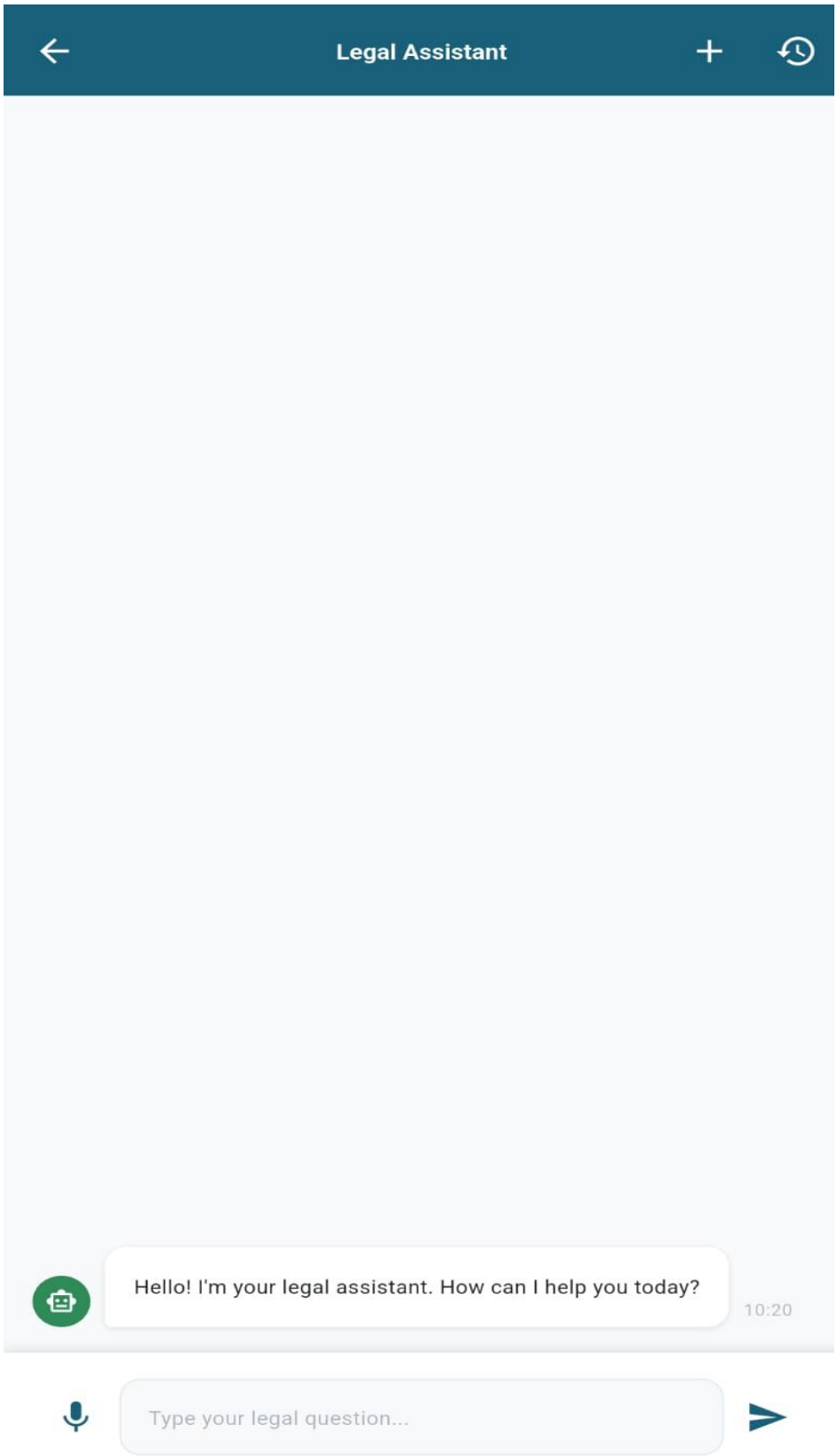

Figure 13 AI Assistant screen of the SmartLaw

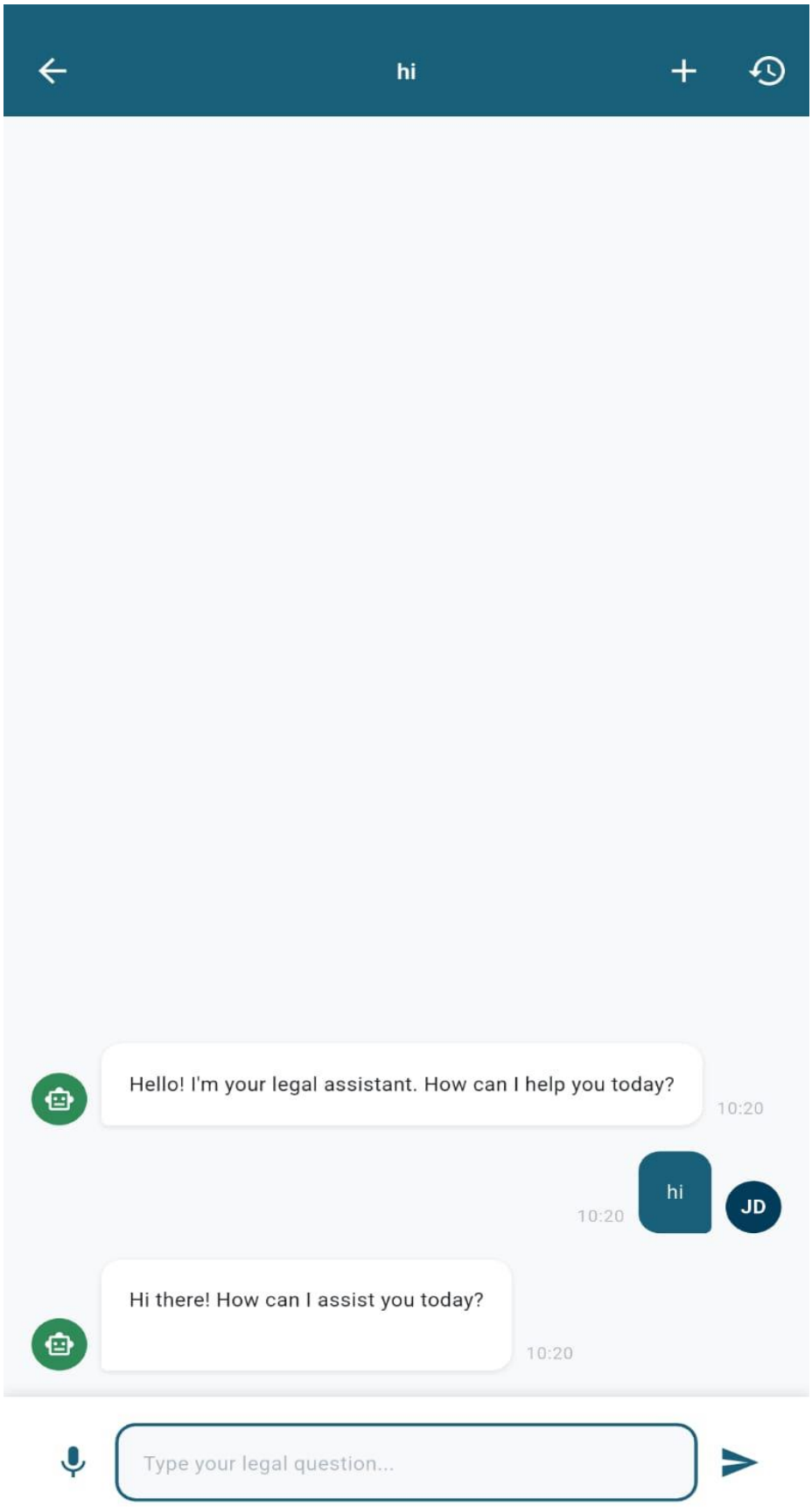

Figure 14 AI Assistant screen of the SmartLaw

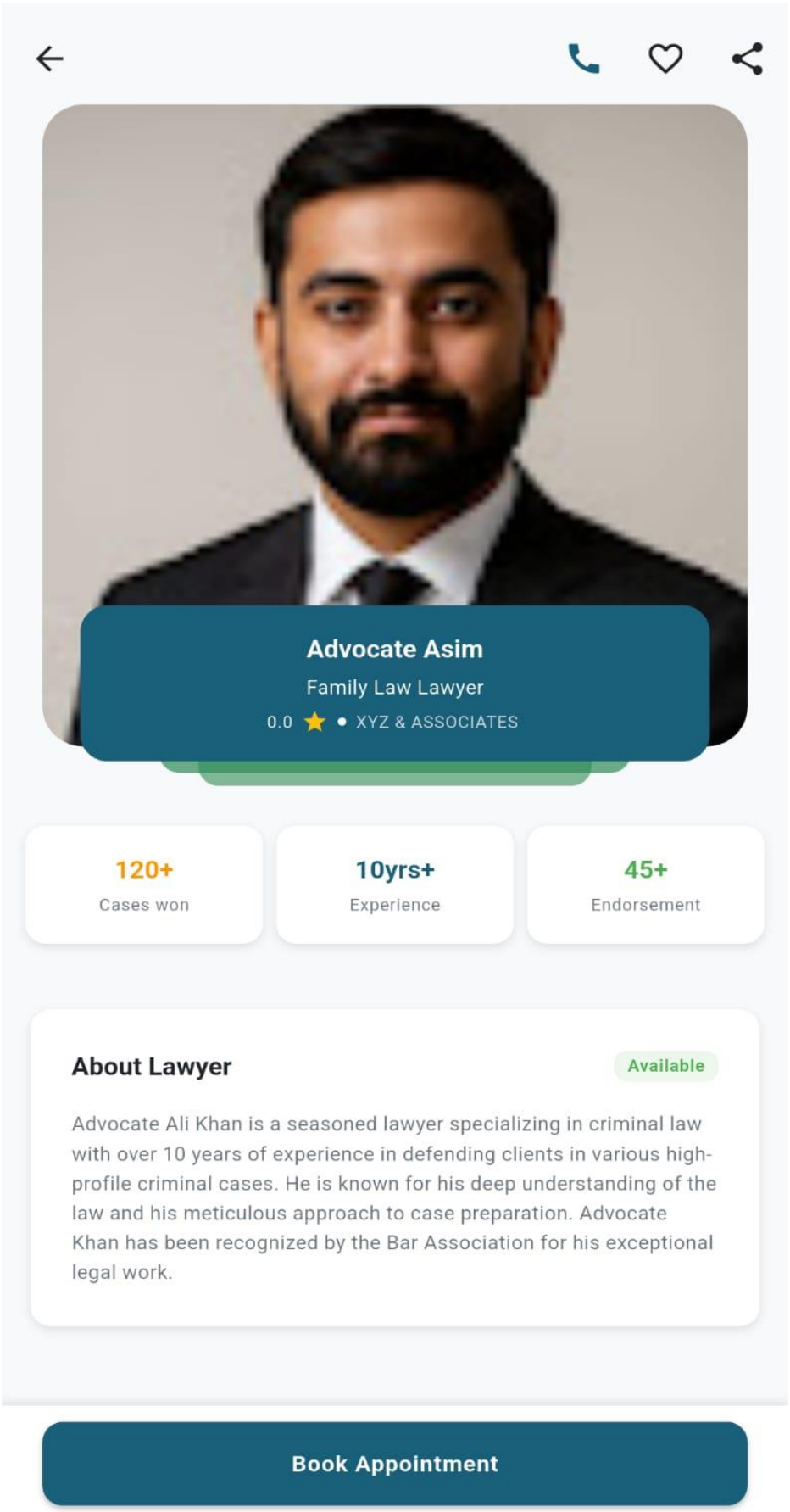

Figure 15 Lawyer's Profile screen of the SmartLaw

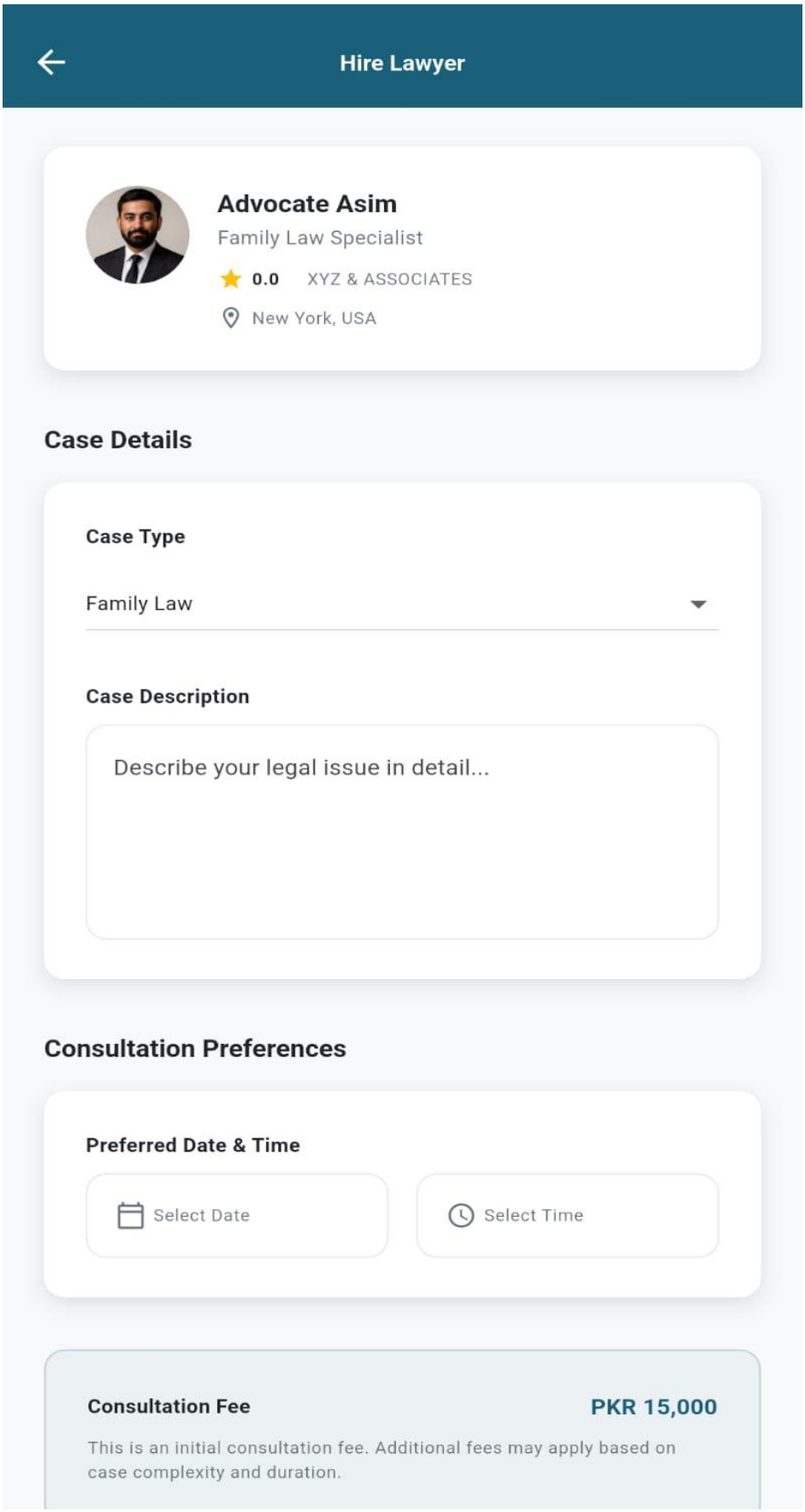

Figure 16 Hiring screen of the SmartLaw

6.2 Performance Evaluation

The SmartLAW system demonstrated reliable performance in identifying relevant criminal law sections based on user queries. Through experimentation, models such as BERT and LSTM were observed to effectively interpret and classify legal statements with a high degree of accuracy.

- Section Mapping Accuracy: 87%
- F1-Score: 92%
- Chatbot Query Success Rate: 85% (on first attempt)
- User Testing Feedback: Positive overall, with emphasis on enhancing interpretability

These results indicate SmartLAW's ability to serve as a viable legal recommendation engine in the domain of criminal law.

6.3 Challenges Faced

6.3.1 Ambiguity in Natural Language

One of the most significant challenges was handling vague or context-lacking user inputs. For instance:

"He threatened me at night."

This could relate to Section 506 (criminal intimidation) or possibly even Section 511 (attempt to commit offense), depending on intent and action details. The AI had difficulty resolving such ambiguity without contextual enrichment.

6.3.2 Multilingual Input Handling

Due to SmartLAW's support for both English and Urdu, language preprocessing became complex. Code-switching and Roman Urdu further added difficulty in model understanding, requiring additional normalization and tokenization logic.

6.3.3 Limited Dataset Scope

While the dataset was based on PPC sections and synthetic cases, a lack of access to real criminal case data (e.g., FIRs, court verdicts) limited the variety of training samples. This somewhat affected the model's generalization to highly nuanced or rare criminal scenarios.

6.3.4 Legal and Ethical Considerations

Given the sensitivity of criminal law, several legal and ethical factors were prioritized during development:

- Privacy Assurance: No user data is stored or logged.
- Advisory Disclaimer: The tool provides assistance, not legally binding interpretations.
- Bias Minimization: Class balance and neutral phrasing were ensured during data collection and augmentation.
- Lawyer Oversight: Legal experts reviewed outputs to prevent misinformation.

These steps ensured SmartLAW operates within the ethical boundaries required for a system dealing with legal matters.

6.4 Use Cases and Societal Impact

SmartLAW serves multiple user segments:

- Individuals: Understand the implications of actions or incidents from a criminal law perspective
- Students: Reference legal sections for study and mock case analysis
- Junior Lawyers: Quickly find relevant PPC sections for case prep
- NGOs & Law Clinics: Educate underserved communities

The chatbot's ability to explain criminal laws in layman-friendly language enhances access to justice and civic awareness.

6.5 Limitations

Despite its success, SmartLAW has some notable limitations:

- Static Legal Database: No live updates for amendments or recent rulings
- No Case History Memory: Stateless chatbot responses lack long-term context
- No Legal Representation Linkage: The tool does not yet connect users to real lawyers or legal aid

These issues do not hinder core functionality but do limit the system's depth in real-world legal scenarios.

6.6 Lessons Learned

- User Query Design Matters: Small variations in phrasing can significantly affect AI understanding.
- Legal NLP Requires Precision: General-purpose models struggle with legal nuance without fine-tuning.
- Domain Collaboration is Crucial: Feedback from legal professionals helped fine-tune system accuracy and language clarity.
- Criminal Law is Multi-Layered: Real-life cases depend heavily on evidence, context, and judiciary interpretation – not just textual law sections

CHAPTER 7

FINDINGS AND RECOMMENDATIONS

7.1 Conclusion

Smart Law demonstrates that AI-powered legal assistance tailored to the Pakistan Penal Code (PPC) is not only technically feasible but also socially impactful. By combining natural language processing (NLP) with a bilingual mobile interface, the system bridges the gap between complex legal frameworks and the layperson, enabling users—especially those unfamiliar with legal terminology or without access to legal counsel—to obtain preliminary legal guidance with ease.

The application of GPT-based AI models enables intelligent, context-aware legal recommendations based on user input, ensuring relevance and accuracy within the bounds of Pakistani criminal law. Furthermore, the use of Flutter for cross-platform mobile development and Firebase for backend support ensures the solution is both scalable and maintainable, with the ability to reach a wide demographic regardless of geographic or socio-economic barriers.

Beyond its technical strengths, Smart Law contributes significantly to the democratization of legal awareness. It empowers citizens to understand their rights, recognize legal violations, and seek timely action without depending on intermediaries or expensive consultations. This is especially critical in rural areas and underprivileged communities, where access to professional legal advice is often limited or non-existent.

The project also sets a precedent for future legal-tech innovations in Pakistan. With modular AI integration and cloud-based architecture, Smart Law can serve as a foundation for further developments in digital justice systems, legal education tools, and even formal judicial support systems.

In essence, Smart Law is not just a mobile application—it is a step toward inclusive legal empowerment, accessible justice, and AI-driven reform in the Pakistani legal landscape.

7.2 Future work

To build upon the current foundation and improve system capability, the following enhancements are proposed for future development:

- **Web Version Deployment**
  Develop and launch a responsive web application to complement the mobile app, ensuring broader accessibility on all platforms including desktop and tablet.

- **FIR Status Integration**
  Integrate real-time FIR status APIs from regional police departments (e.g., Punjab Police, Islamabad Police) to allow users to check the status of filed FIRs directly within the app.

- **Legal Document Validation**
  Implement an AI-powered document validator to authenticate and assess the correctness of legal documents such as affidavits, rent agreements, or contracts.

- **Expanded Legal Domains**
  Extend legal recommendations beyond criminal law to include civil law, family law, labor law, and property law, increasing the system's utility for a broader range of legal scenarios.

- **Admin Portal for Case Monitoring (Planned)**
  Introduce an admin panel to monitor, manage, and improve the accuracy of AI recommendations and user engagement metrics.

- **Offline Mode Support**
  Add offline capabilities to allow users in remote areas to fill forms and receive basic guidance even without a live internet connection.

- **Legal Awareness Features**
  Integrate educational modules or FAQs to raise awareness of basic rights under Pakistani law, especially among non-technical users.

REFERENCES

[1] D. L. Carter, "Brief history of law enforcement intelligence: Past practice and recommendations for change," *Trends Organ. Crime*, vol. 8, no. 3, pp. 51–62, 2005.

[2] R. Brooks and C. Lopez, "Recommendations for law enforcement," *Justiceroundtable.org*. [Online]. Available: https://justiceroundtable.org/wp-content/uploads/2020/04/whitepaper7a.pdf.

[3] M. R. Tahouni, E. Liscord, and H. Mowafi, "Managing law enforcement presence in the emergency department: Highlighting the need for new policy recommendations," *J. Emerg. Med.*, vol. 49, no. 4, pp. 523–529, 2015.

[4] M. A. Tayebi, M. Jamali, M. Ester, U. Glässer, and R. Frank, "CrimeWalker: A recommendation model for suspect investigation," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 173–180.

[5] *Gov.pk*. [Online]. Available: http://www.ljcp.gov.pk/nljcp/assets/dist/news_pdf/aji.pdf.

APPENDICES

APPENDIX A: Code

```
String _extractResponseText(String responseBody) {
  try {
    final jsonResponse = json.decode(responseBody);

    if (jsonResponse is Map) {
      // Check for common response formats
      for (final key in ['response', 'text', 'message', 'content']) {
        if (jsonResponse.containsKey(key) && jsonResponse[key] is String) {
          return jsonResponse[key];
        }
      }
    }

    // If it's just a JSON string or we couldn't find a known key
    return jsonResponse is String ? jsonResponse : responseBody;
  } catch (_) {
    // If not valid JSON, return the raw response
    return responseBody;
  }
}
```

Figure 17 AI Chatbot code screenshot

```
Future<void> _sendMessageToAPI(String message) async {
  final baseUrl = dotenv.env['API_BASE_URL'];
  final url = Uri.parse('$baseUrl/api/chat');

  try {
    // Prepare payload - Send message with current chat history
    final payload = {
      'prompt': message,
      'chatHistory': _chatHistory,
    };

    // Send the request
    final response = await http.post(
      url,
      headers: {'Content-Type': 'application/json'},
      body: json.encode(payload),
    );

    // If response is successful
    if (response.statusCode == 200) {
      final modelResponse = _extractResponseText(response.body);

      // Update chat history with the conversation pair
      _chatHistory.add({
        "role": "user",
        "parts": [
          {"text": message}
        ]
      });

      _chatHistory.add({
        "role": "model",
        "parts": [
          {"text": modelResponse}
        ]
```

Figure 18 AI Chatbot code screenshot

```
// Determine if current user is a lawyer or regular user
final bool currentUserIsLawyer = await _checkIfUserIsLawyer(currentUserId);

// Collections for the current user and the chat partner
final String currentUserCollection =
    currentUserIsLawyer ? 'lawyers' : 'users';
final String partnerCollection = currentUserIsLawyer ? 'users' : 'lawyers';

try {
  // First check if partner exists in the expected collection
  final partnerDoc = await FirebaseFirestore.instance
      .collection(partnerCollection)
      .doc(partnerId)
      .get();
```

Figure 19 AI user lawyer chat code screenshot

```dart
// Generate a unique chat ID
final String chatId = _getChatId(currentUserId, partnerId);

// Create the message data
final messageData = {
  'text': text,
  'timestamp': FieldValue.serverTimestamp(),
  'senderId': currentUserId,
  'senderName': userData.data()?['name'] ?? 'Unknown',
  'senderImage': userData.data()?['photoUrl'] ?? '',
  'receiverId': partnerId,
};

// 1. First add the message to the main chats collection
final messageRef = await FirebaseFirestore.instance
    .collection('chats')
    .doc(chatId)
    .collection('messages')
    .add(messageData);

final String messageId = messageRef.id;

// 2. Save the message in the current user's collection
await FirebaseFirestore.instance
    .collection(currentUserCollection)
    .doc(currentUserId)
    .collection('chat')
    .doc(partnerId)
    .collection('messages')
    .doc(messageId)
    .set(messageData);
```

Figure 20 AI user lawyer chat code screenshot