Problem Statement: Analysis of FIR using AI/ML for proper Act and Section.

Problem Statement:
To build an AI/ML system so that stronger sections of the law can be applied to the FIR being registered and the sections applied are relevant to accusations and the investigations that follow.

Solution:
1. **OCR Functionality:**
   Using the powerful Google Tesseract engine we can extract the handwritten text from a document. Google Tesseract is a powerful OCR library which can easily recognise text.
2. **State-specific Rules:**
   By using custom-trained state-of-the-art Large Language Models we can make sure that State-specific local rules alongside CrPC and IPC should be incorporated for accurate section recommendations.
3. **User Flexibility:**
   The prompt given to LLM is custom in nature and can be used to modify the course of action taken by the model while writing the FIR.
4. **Speed and Accuracy:**
   LLMs are highly capable models which can summarize a large investigative report and provide a summarized version or a full FIR in a matter of seconds. This will help the investigating officers by reducing the time to research the relevant sections and creating a full document from scratch.
5. **Standardization:**
   We can use a uniform process for consistency by implementing a standardization protocol for text extraction and data analysis.
6. **Transparency and Accountability:**
   Maintaining a transparent process holding the system accountable for the sections applied and resource-saving**.**
7. **Compatibility with CCTNS:**
   Ensure the system is compatible with the Crime and Criminal Tracking Network & Systems (CCTNS) for seamless integration by developing APIs for integration with CCTNS.
8. **Law Application Highlighting:**
   Portal displaying properly maintained data after the systematic analysis of Statement, Highlighting all the applicable law sections for each statement or paragraph, aiding in training. Portal includes text search, document scanning and speech analysis through NLP.
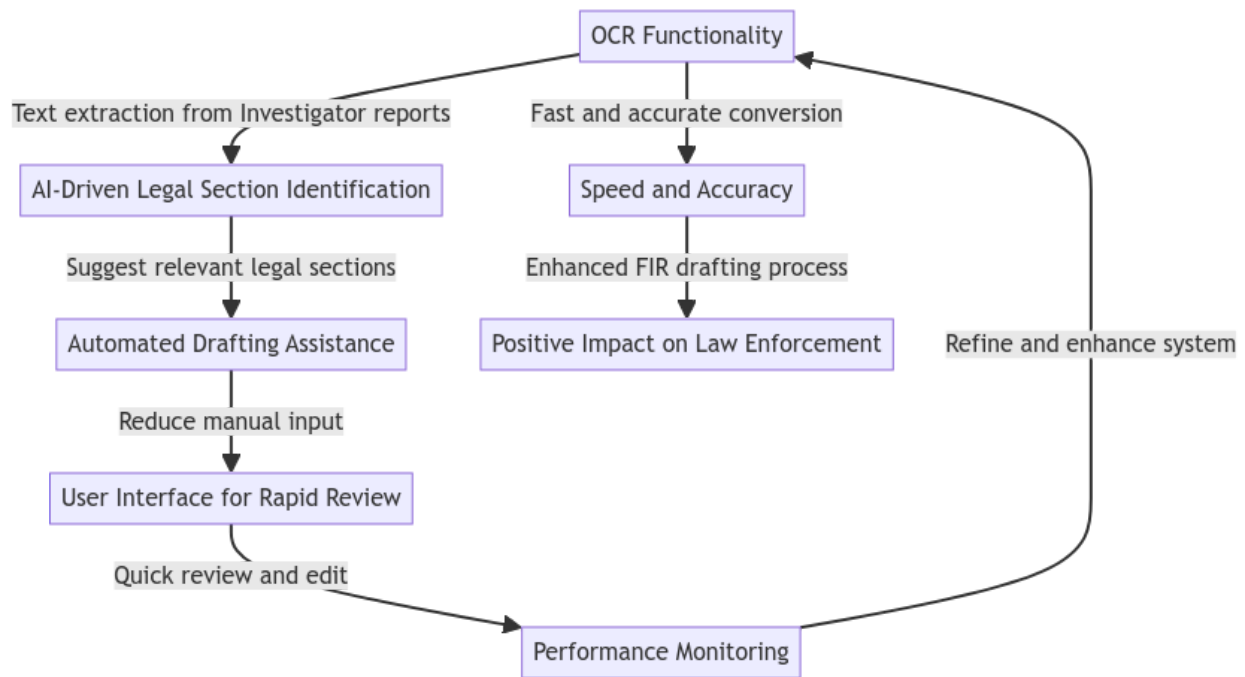
Software and Technology:

- **Languages:** Python, HTML, JavaScript, CSS
- **Framework:** Django, React, PyTorch, TensorFlow, PyTesseract
- **Libraries:** Bootstrap/Tailwind
- Mobile Development: Flutter

Team Members & Responsibilities:

● Arun Kushwaha - Frontend Development

- Chaitanya Kohli - ML model
- Ananay Kaushal- ML model and backend development
- Vishesh Garg- Backend Development

Flow Chart / Graphical Representation:

```
                                    ┌──────────────────┐
                                    │ OCR Functionality│
                                    └──────────────────┘
              Text extraction from            Fast and accurate
              Investigator reports            conversion
                      │                             │
                      ▼                             ▼
        ┌───────────────────────────┐    ┌──────────────────┐
        │ AI-Driven Legal Section   │    │ Speed and Accuracy│
        │ Identification            │    └──────────────────┘
        └───────────────────────────┘             │
              Suggest relevant                Enhanced FIR
              legal sections                  drafting process
                      │                             │
                      ▼                             ▼
        ┌───────────────────────────┐    ┌───────────────────────────┐
        │ Automated Drafting        │    │ Positive Impact on Law    │    Refine and enhance system
        │ Assistance                │    │ Enforcement               │
        └───────────────────────────┘    └───────────────────────────┘
              Reduce manual input
                      │
                      ▼
        ┌───────────────────────────┐
        │ User Interface for Rapid  │
        │ Review                    │
        └───────────────────────────┘
              Quick review and edit
                      │
                      ▼
              ┌───────────────────────┐
              │ Performance Monitoring│
              └───────────────────────┘
```

Schedule:

Step 1: Frontend of the portal.
Step 2: Prepare/Gather required dataset.
Step 3: Train ML Model.
Step 4: Integration of CCTNS with the model.
Step 5: Backend of the portal.
Step 6: Integration of backend and frontend.