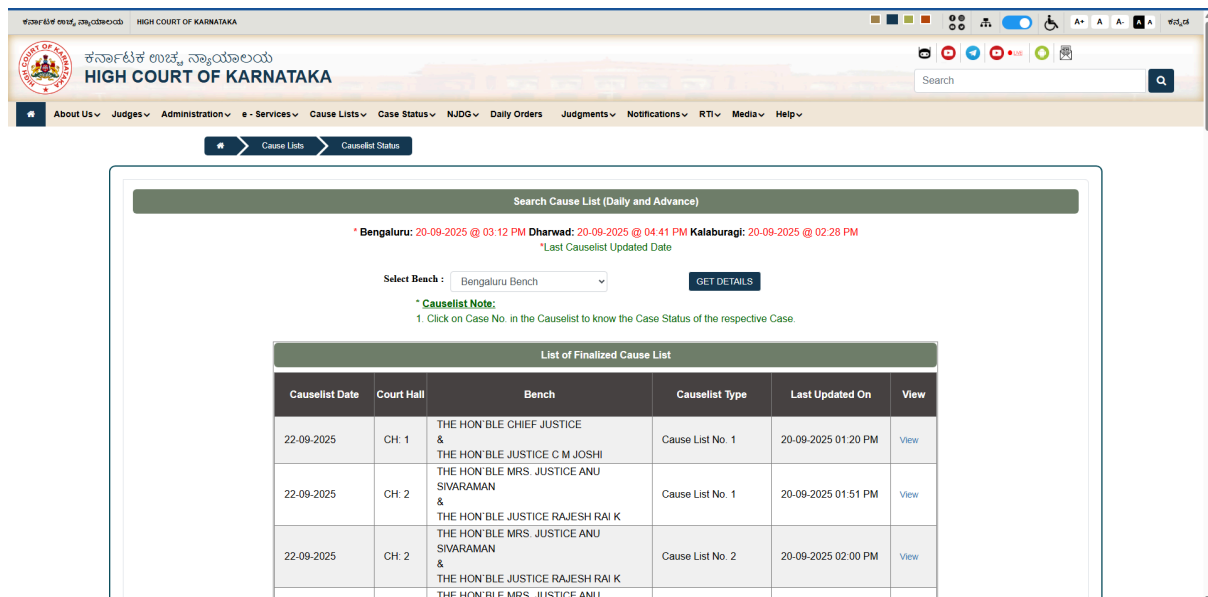


Hackathon Problem Statement: Smarter Court Scheduling for Justice Delivery

For Working Professionals

Introduction

Case scheduling - known as “listing in the judiciary - is a critical yet poorly understood function of court administration. Each day, courts release [cause lists](#) that decide which cases will be heard, in what order, and before which bench. This administrative process has profound consequences: it determines how quickly disputes are resolved, whether urgent matters receive priority, and how judicial time is allocated.



Today, scheduling in Indian courts suffers from unpredictability, high adjournment rates, and uneven workload distribution. Judges face packed dockets, lawyers push for urgent mentions, and litigants often endure long waiting periods. Despite the existence of digital case management systems, the logic behind scheduling remains opaque, fragmented, and often reactive – as well as being highly variable and personality-dependent.

By working with a large dataset of over **134,699 unique civil cases and 740K hearings** from the Karnataka High Court, participants will confront one of the judiciary’s hardest challenges: how to design fair, efficient, and adaptable scheduling systems at scale.

Objectives

The hackathon is designed with four core objectives:

1. Draw Comparative Insights

Examine how scheduling and optimisation challenges are solved in other complex systems such as healthcare (surgery rosters), airlines (crew and flight scheduling), or logistics (supply chain optimisation) and extract lessons that can be adapted to the judicial context.

Participants are encouraged to leverage the knowledge and experience they have gained from their own work areas, applying familiar strategies and problem-solving approaches to this new setting. This

cross-domain thinking helps generate creative, practical solutions that address court scheduling and workflow challenges effectively.

2. Understand Case Progression and Data Patterns

- Use real-world court data to understand how cases move across stages in the lifecycle of a case and the different purposes of court hearings: admission, evidence, arguments, interlocutory applications, and disposal.
- Identify bottlenecks, variations in listing frequency, and signals of case readiness

3. Develop a Predictive Scheduling Algorithm

- Design an algorithm that balances fairness, urgency, and efficiency while respecting judicial constraints such as bench composition, leaves, limited daily time, and discretion.
- Incorporate flexibility for cases requiring urgent relief, adjournments, and judge-specific preferences.

4. Prototype a Practical Tool

- Build a functional prototype that can simulate daily court scheduling, generating daily cause lists¹ for multiple courtrooms.
- Demonstrate usability, adaptability, and clear documentation of logic and assumptions.

Approach

The hackathon challenge will include three steps for the teams to follow:

1. Step 1 – Comparative Insights

Participants will briefly document how other industries solve complex scheduling problems under constraints. This would ideally draw on their professional experience across sectors and make explicit the assumptions, necessary data, and model conditions they are using for the judicial context.

For example:

- In healthcare, any surgeries from the daily ‘[surgery board](#)’ that are not reached are re-prioritised based on urgency and resource availability.
- In airlines, flight scheduling balances crew availability, passenger demand, and aircraft turnaround.
- In manufacturing, production scheduling optimises throughput while managing bottlenecks.

The goal is to identify principles that can inspire judicial scheduling reforms.

2. Step 2 – Data-Informed Modelling

Using the provided dataset of historical court data, participants will:

- Analyse case timelines, hearing frequencies, and listing patterns.

¹ These daily lists can be designated by ‘assigning’ dates in a compiled dataset – there is no need to generate daily PDFs or specific formats for the lists.

- Determine how cases could be classified as “ripe” or “unripe” based on purposes of hearing and stage. *Real-world data shows that cases can be frequently listed but if there is a bottleneck (such as summons not being received or a case that cannot progress until another court delivers a verdict), then merely scheduling the case for a hearing does not move towards disposal. Participants can consider various factors that would require scheduling in a case to be paused and recommend ways to mark such cases and update the scheduler when they are ready to be listed again*
- Develop assumptions about new case inflows, adjournments, and weekly disposal rates.
- Identify gaps in current data capture and propose synthetic fields or data enrichment strategies that could be adopted by courts.

3. Step 3 – Algorithm Development

Participants will design and implement a scheduling algorithm that:

- Allocates cases dynamically across multiple simulated courtrooms.
- Accounts for judicial working days, time limits, and caseload distribution.
- Generates daily cause lists with room for supplementary additions by judges.
- Simulates case progression over a two-year period, ensuring no case is left behind.

Expected Outputs

At the end of the hackathon, participants are expected to deliver:

- **A brief** (written document, video, or presentation) mapping best practices in scheduling and optimisation from other sectors, stating the assumptions or logic they have adopted, covering analytical insights into data quality gaps, and recommendations for improving case metadata capture in the future.
- **An algorithmic model** capable of simulating case allocation and progression under dynamic, real-world-like conditions (should be shared as a functional link accessible to evaluators or a video walkthrough).
- **A simulated dataset containing cause lists generated by the prototype** that contains daily schedules for at least 5 courtrooms, incorporating judicial discretion and urgent flexibility.

Note: Based on the tools and programming languages selected, the specific formats of outputs may vary. An example of specific output could be: (1) Written brief with assumptions, scenario parameters, metrics report (such as average frequency of hearing or disposal timeline), (2) a 3-minute narrated video with the running model, and (3) a csv file containing the dataset - including simulated dates of hearing.

Potential Outcomes

The outcomes of this hackathon go far beyond the competition itself. If successful, solutions can:

- **Transform court efficiency** by reducing adjournments, improving predictability, and ensuring long-pending cases don’t fall through the cracks.
- **Empower judges** with intelligent scheduling support tools that balance their workload while preserving independence and discretion.

- **Enhance litigant trust** by making hearings more timely and transparent, reducing frustration caused by delays.
- **Provide scalable models** that can be adapted across High Courts, lower courts, and tribunals in India and globally.

Evaluation Criteria

Participants' submissions will be assessed based on:

- **Robustness of Logic and Assumptions:** How well the algorithm is grounded in sound reasoning, and how clearly assumptions are defined and justified.
- **Applicability to Real-World Conditions:** The algorithm's effectiveness in handling dynamic, imperfect, and unpredictable conditions in judicial scheduling.
- **Clarity and Transparency of Documentation:** How comprehensively the methodology, assumptions, and decision-making process are documented for reviewers and potential implementers.
- **Balancing Efficiency with Equity:** The ability to optimise scheduling while considering fairness, discretion, and practical constraints in real-world court operations.

This challenge asks participants not just to code but to think like reformers, operations managers, and system designers. The task is to imagine a future where court schedules are not a source of frustration but a driver of justice.