

EPIC 1 – Universal Date & Time Intelligence Engine

Epic Description

Build a robust NLP-based engine capable of detecting, parsing, and normalizing date and time expressions in multiple formats, languages (English/Hindi/Hinglish), and permutations.

This engine must handle both structured and unstructured date patterns.

USER STORY 1.1 – Multi-Format Date Recognition Engine

As a system,

I want to detect date references in multiple natural formats and permutations, so that users can type dates freely.

In-Scope

System must detect:

Numeric Formats

- 01/12/2026
- 1/12/26
- 1-12-2026
- 2026-12-01
- 1 12 2026
- 12.01.2026

Text Formats

- 1 Jan
- Jan 1
- 1 January 2026
- January 1st
- 1st Jan

- March 2026
- Feb 18

✓ Relative Dates

- Today
- Tomorrow
- Kal
- Parso
- Next Monday
- This Friday
- Next month 5th

✓ Mixed Language

- Kal 5 pm
 - 1 Jan ko meeting
 - March 2026 mein
-

Out-of-Scope

- Lunar/Islamic calendar support (Phase 2)
 - Historical ambiguous dates older than 1900
-

Acceptance Criteria

- AC1: $\geq 95\%$ correct extraction on test dataset
 - AC2: Supports at least 30+ date permutations
 - AC3: Resolves relative dates using timezone
 - AC4: Handles ambiguous format (1/12/2026) using locale rule
-

Validation Rules

- Year range: 1900–2100
 - If year missing → assume current year
 - If month missing → ask clarification
 - If ambiguous numeric (1/12/2026):
 - India locale → DD/MM/YYYY
 - US locale → MM/DD/YYYY
-

Error Messages

- “I found a date but need clarification: Is this 1 December or 12 January?”
 - “Invalid date format.”
-

Definition of Done

- Date pattern library created
 - Regex + NLP hybrid engine implemented
 - Locale-aware parser
 - 500+ test cases validated
 - Unit test coverage >90%
-



USER STORY 1.2 – Time Recognition & Normalization

As a system,

I want to detect multiple time formats and convert them into 24-hour standard format.

In-Scope

Detect:

- 5 pm
 - 5PM
 - 17:00
 - 5:30 am
 - 0530
 - 5 baja
 - 5 shaam
 - Morning
 - Evening
 - Afternoon
 - Night
-

Default Logic

- Morning → 09:00

- Afternoon → 14:00
 - Evening → 18:00
 - Night → 21:00
-

Acceptance Criteria

- ≥ 95% detection accuracy
 - Converts all times to HH:MM (24-hour)
-

Validation Rules

- 00:00–23:59 only
 - If time missing → ask user or default 09:00
-

Error Message

- “Invalid time detected.”
-

DoD

- Time parser implemented
 - Relative time logic tested
 - Edge case coverage
-

EPIC 2 – Date-Time Normalization & Structured Event Engine

Epic Description

Normalize extracted date-time expressions into a standardized internal structure for system-wide consistency.

USER STORY 2.1 – Flexible Event Object Builder

As a system,

I want to convert extracted data into a normalized event object.

Required Internal Format (Flexible, Not Mandatory Fixed JSON)

System should support:

```
event_id
event_title
start_datetime (ISO 8601)
end_datetime (optional)
timezone
confidence_score
source (chat/email)
raw_text
```

Acceptance Criteria

- ISO 8601 conversion (e.g., 2026-01-01T17:00:00+05:30)
 - Timezone resolved
 - Raw input stored for audit
-

Validation Rules

- Start datetime must not be in the past
 - Title required (if missing → generate from context)
-

Error Message

- “Event time is in the past.”
 - “Event title missing. Please confirm.”
-

DoD

- Normalization API deployed
- Handles single & multiple dates

- Supports partial input completion
-

EPIC 3 – Smart Clickable Date Highlight System

Epic Description

Make detected date-time entities clickable within chat/email preview.

USER STORY 3.1 – Dynamic Date Highlight UI

As a user,

I want detected date/time text to be highlighted and clickable.

In-Scope

- Underline or color highlight
 - Click action popup:
 - Add to Calendar
 - Add Reminder
 - Edit
-

Acceptance Criteria

- Click triggers structured preview modal
 - No false positives >5%
-

Validation Rules

- Highlight only high-confidence entities (>80%)
-

Error Message

- “Unable to process this date.”
-

DoD

- Frontend integration complete
 - Works in chat & email preview
-

EPIC 4 – Native Calendar & Reminder Integration

Epic Description

Enable event creation in native apps via deep linking or API integration.



USER STORY 4.1 – Native Calendar Deep Linking

As a user,

I want to add events to my default calendar app instantly.

In-Scope

- Android (Google Calendar)
 - iOS (Apple Calendar)
 - Pre-fill:
 - Title
 - Date
 - Time
 - Notes
-

Acceptance Criteria

- Opens default calendar
 - Fields correctly pre-filled
 - Permission handling implemented
-

Error Messages

- “Calendar permission not granted.”
 - “No calendar app found.”
-

DoD

- Android & iOS tested
 - Edge cases handled
-

EPIC 5 – Optional In-App Reminder System (Phase 2 Exploration)

Epic Description

Build internal reminder engine with push notifications.

USER STORY 5.1 – Identity-Based Reminder Storage

As a user,

I want reminders saved in Samay app so they sync across devices.

In-Scope

- Cloud storage
 - Push notifications
 - Snooze feature
-

Out-of-Scope (MVP)

- Recurring complex rules
 - Shared reminders
-

EPIC 6 – Analytics & Accuracy Monitoring

Epic Description

Monitor extraction performance & usage behavior.

Metrics to Track

- % messages with date/time
 - Extraction accuracy
 - Click-to-create ratio
 - Failure rate
 - Most used formats
-

DoD

- Dashboard deployed
 - Confidence tracking active
 - Alert if accuracy <90%
-

Implementation Order

1. Date Permutation Engine
2. Time Detection Engine
3. Locale-Aware Resolver
4. Event Normalization API
5. Clickable Highlight UI
6. Native Calendar Deep Linking
7. Analytics

8. In-App Reminder (Optional)

Technical Recommendation for AI Team

Use Hybrid Approach:

- Regex library for numeric formats
 - NLP entity recognition model (NER)
 - Locale-based resolver
 - Confidence scoring
 - LLM fallback for ambiguous phrases
-