EPICS

EPIC 1: Ticket Creation & Categorization

Goal: Allow customers or support agents to raise new tickets, capturing all relevant details.

User Stories:

- **US 1.1**: As a *customer*, I want to raise a ticket via mobile/web with issue details, so that I can report broadband issues easily.
- **US 1.2**: As a *support agent*, I want to log a ticket on behalf of the customer calling in, so that manual channels are also covered.
- **US 1.3**: As the *system*, I want to auto-categorize the ticket based on keywords, so that triaging can be faster.
- **US 1.4**: As the *system*, I want to assign a unique Ticket ID and timestamp when the ticket is created.

EPIC 2: Ticket Assignment & SLA Tracking

Goal: Route the ticket to the right team and ensure SLA monitoring begins.

User Stories:

- **US 2.1**: As a *triage officer*, I want to assign a severity and priority to tickets based on the description, so that they are handled in the right order.
- **US 2.2**: As the *system*, I want to route tickets to the appropriate team (e.g., Field Engineer or NOC), so that it reaches the correct resolver.
- **US 2.3**: As a *manager*, I want to track SLA countdowns per ticket, so that delays can be escalated.
- **US 2.4**: As a *resolver*, I want to view all open tickets assigned to me, prioritized by SLA urgency.

EPIC 3: Ticket Diagnosis and Resolution

Goal: Support engineers to analyze, act, and resolve network issues effectively.

User Stories:

- **US 3.1**: As a *resolver*, I want to update ticket status (e.g., "In Progress", "On Hold", etc.) as work progresses.
- **US 3.2**: As a *resolver*, I want to record actions taken (e.g., port reset, ONT replaced), so that the resolution history is maintained.
- **US 3.3**: As the *system*, I want to attach logs and diagnostic reports to a ticket, so that technical analysis is retained.
- **US 3.4**: As a *field engineer*, I want to update the ticket via mobile after on-site resolution, even offline if needed.

EPIC 4: Ticket Closure and Feedback

Goal: Ensure verified closure of issues and record user satisfaction.

User Stories:

- **US 4.1**: As a *resolver*, I want to mark the ticket as resolved once the issue is fixed.
- **US 4.2**: As the *system*, I want to notify the customer with a resolution summary and ask for confirmation.
- **US 4.3**: As a *customer*, I want to confirm that the issue is resolved, or reopen if not.

- **US 4.4**: As a *customer*, I want to provide feedback or rate the service after ticket closure.

EPIC 5: Dashboard, Reports & Alerts

Goal: Monitor ticket trends, SLA breaches, engineer performance.

User Stories:

- **US 5.1**: As a *manager*, I want to view open, resolved, overdue, and escalated tickets by team/location.
- **US 5.2**: As a *team lead*, I want to be alerted if SLA for any ticket is about to breach.
- **US 5.3**: As a *NOC admin*, I want to view ticket heatmaps for frequent outage zones.
- **US 5.4**: As a *CXO*, I want daily or weekly reports on ticket volume, resolution time, and satisfaction scores.

EPIC 6: AI-Powered Support & Recommendations

Goal: Use Gen AI to assist in triaging and suggesting solutions.

User Stories:

- **US 6.1**: As a *support agent*, I want AI to suggest severity and probable root cause based on user input.
- **US 6.2**: As a *resolver*, I want to ask AI for possible resolution steps based on ticket history and description.
- **US 6.3**: As the *system*, I want to auto-categorize and even auto-resolve known simple issues (e.g., reboot instructions).
- **US 6.4**: As a *customer*, I want to interact with a chatbot that can guide me through common fixes before logging a ticket.