



Agile Implementation – Sahayog Sarathi (Kanban Method)

Nepal's digital landscape needs a localized crowdfunding solution. Global platforms often miss the mark on payment systems and cultural nuances. Our answer is **Sahayog Sarathi** – a transparent, secure, and user-friendly Nepali crowdfunding platform, built to support local languages, regional payment systems, and foster community trust.

Team Roles & Dynamics

Meet the Team



Ashik Chapagain

Backend Developer: Campaign APIs, fund withdrawal, payment integrations, admin tools.



Anupam Sangam

Frontend Developer: UI components, dashboard, campaign creation page, responsive design.



Prabin Kandel

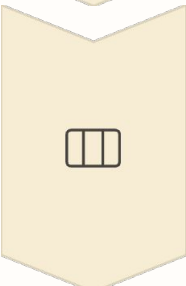
QA: Test cases, usability testing, quality assurance.

Kanban in Action



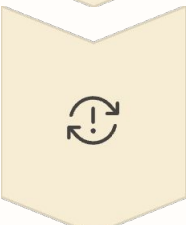
Visual Task Tracking

Utilized Trello for clear task visualization.



Defined Workflow

Tasks moved through To Do, In Progress, Review, and Done columns.



Continuous Flow

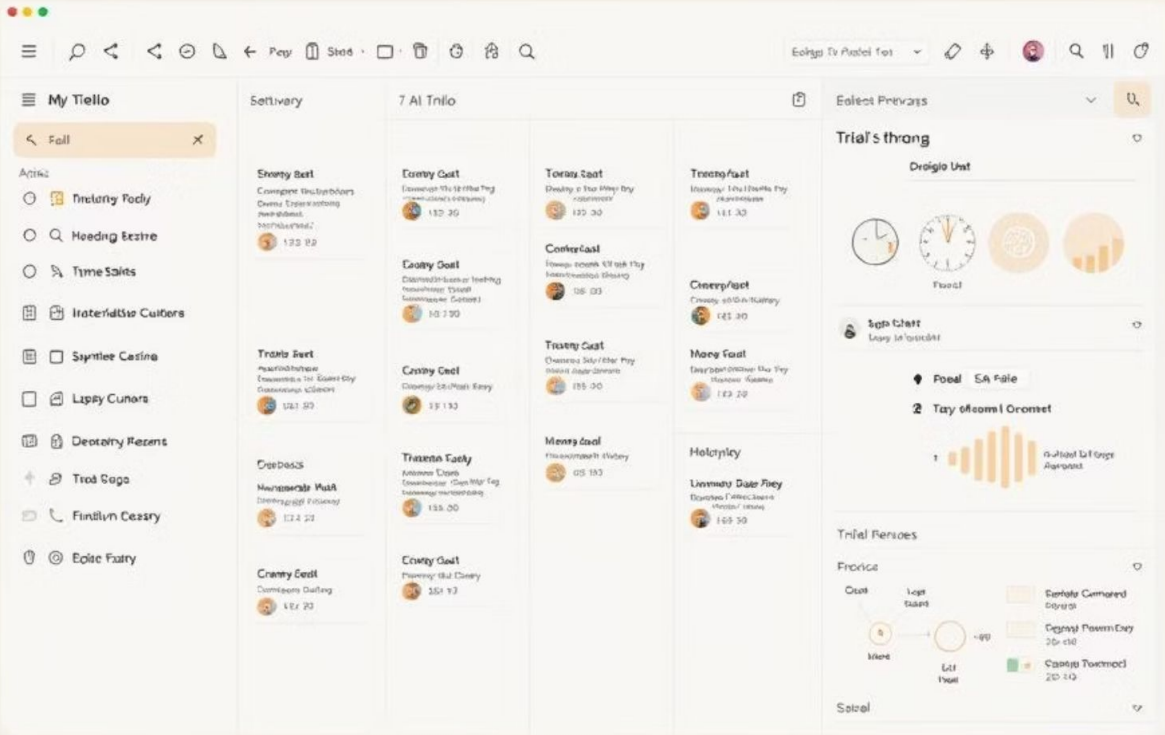
Enabled flexible, continuous updates over rigid sprints.



Self-Organizing

Team members pulled tasks based on expertise and availability.

Planning Process & Backlog Creation



Tools that Drove Our Planning

Trello

Kanban board for task tracking and progress visualization.

Figma

UI/UX wireframing, mockups, and detailed flow planning.

Draw.io

Database ER diagrams and user journey mapping for clarity.

GitHub

Centralized version control and seamless team collaboration.

Strategic Backlog Creation

- Features as Cards: Each platform feature became a detailed card.
- Priority Tagging: Cards were tagged High/Medium/Low priority.
- Progress Updates: Checklists and comments tracked real-time progress.
- Traceability: GitHub commits linked directly to Trello cards.




Demonstration of Progress

Real-time Task Progression

Our Kanban board provided instant visibility into task statuses, fostering transparency and accountability.

Seamless Task Movement

"Integrate Khalti payment" –  Done (Ashik)

"Campaign gallery layout" –  In Review (Anupam)

"Verification system with document upload" –  In To Do

Empowered Team Updates

Every team member owned the responsibility of updating their task status, ensuring accurate and timely progress reporting.

Bottleneck Prevention

Kanban's asynchronous nature allowed parallel work streams, effectively preventing bottlenecks and maintaining a smooth workflow.

Retrospective Learnings



What Went Well

Visual Clarity & Flow Control

Kanban provided an immediate, clear overview of the project flow.

Smooth Parallel Development

Backend APIs and frontend UI progressed independently, minimizing dependencies.

Flexible Reprioritization

Easy adaptation to changing priorities or new blockers.

Key Improvements During the Process

Card Templates: Introduced standardized templates with detailed descriptions and clear acceptance criteria.

Due Dates & Labels: Added these for critical, time-sensitive features like payment integrations.

Daily Syncs: Utilized Slack and Trello comments for consistent team alignment and communication.

Next Steps & Future Enhancements



Rich Campaign Storytelling Editor

Empowering users with advanced tools for compelling campaign narratives.



Real-time Donation Activity Feed

Creating dynamic engagement with live updates on contributions.



Interactive Progress Bar Animation

Visualizing goal completion with engaging and responsive animations.



SMS Notification Integration

Expanding outreach to rural areas through accessible communication.



Mobile Performance Optimization

Ensuring seamless experience for users on 2G/3G networks.



Backlog



Add multilingual toggle (Nepali/English)

Create donation leaderboard by region/cause

Implement AI-based campaign suggestions

Partner portal for NGOs to manage sub-campaigns

Real-time SMS updates for rural users

Campaign boost/promotion tools

Add blog/newsroom section

Create custom analytics for users

+ Add a card



Todo



Setup PostgreSQL and Prisma ORM



• Integrate eSewa sandbox API



• Create user registration and login page (UI)

• Develop campaign creation form with file upload

• Design admin dashboard UI (Figma or code)

• Implement withdrawal request flow

• Configure NextAuth or Firebase for user sessions

• Build API: create/edit campaign

• Write validation schema for forms

• Setup GitHub repo and project structure

• Configure environment

+ Add a card



In progress



• Implement landing page layout (Next.js + Tailwind)



• Backend API: create and list campaigns



• Build frontend dashboard (My Campaigns)



• Admin view: approve/reject campaigns



• Figma: Campaign detail page (desktop + mobile)



• UI Component: Alert and Notification toast system



• Add Khalti integration with success callback



• Profile page: update avatar, user info



+ Add a card



In Review



• User login system with JWT auth (backend)



• Campaign listing grid (desktop + mobile responsive)



• Integration test: Khalti + eSewa



• Campaign form validation and error messages

• Withdraw request email notifications

• Image upload and preview feature (S3)

• Admin flagging and moderation interface

• Database migration scripts with seed data

+ Add a card



Done



• Trello board setup with labels and priorities

• Project branding (logo, color palette, fonts)

• Setup frontend on Vercel

• Backend deployed to Railway



• GitHub repo initialized with README



• Defined ERD in Draw.io

• Figma high-fidelity mockups for main screens

• Created test accounts (admin, user)



• Payment test flow documented



• Daily standup meeting notes created (if applicable)



+ Add a card



+ Add another list