Contents

Table of Tables

Fundora: Crowdfunding Platform for Nepal

1. 1. Software Requirement Specifications (SRS)

1.1 About SRS

A Software Requirement Specifications (SRS) is a detailed and formal document that defines both functional and non-functional requirements of a software. It outlines the intended purpose and environment for software under development (Barney, 2024). It provides a clear map of the system offering detailed descriptions of the system's features and limitations. Functional requirements deal with what the system is supposed to do, detailing the specific tasks and processes. On the other hand, non-functional requirements define the qualities or the attributes that the system should have, such as high performance, security, user friendly, maintainability, and scalability. SRS ensures that everyone involved in the project from developers to testers and clients get better understanding of the goals and limitations of the system.

1.2 Importance of an SRS in Software Development

A Software Requirement Specification is a crucial document which ensures that all stakeholders have a clear understanding of the project requirements. It plays a key role in guiding the development process and aligning the final product with the client's needs.

The importance of an SRS are listed below:

- Clear Communication: SRS helps to ensure that the developers, clients, and stakeholders have clear communication about the project. It keeps them on the same page regarding the goals and specifications of the project.
- Reduced Development Cost: SRS helps in reducing the risk of costly changes and reworks during the development process. It minimizes errors and misunderstandings, leading to the better utilization of the resources.

- Facilitates Testing and Validation: An SRS serves as a reference for testing and validation. It enables quality assurance teams to come up with better test cases and ensure that the final product meets the specified criteria.
- Improved Project Management: SRS provides a proper basis for project planning and management. It helps in estimating timelines, costs, and resource requirements which leads to better project control.

1.3 Purpose of the Document

The Software Requirements Specification (SRS) offers an extensive overview of the system's requirements, ensuring that all the stakeholders share a mutual comprehension of the functionalities and limitations.

It serves several critical purposes:

- Defines software functionalities and constraints to guide the development process.
- Acts as a reference for the validation and verification of the system.
- Helps manage project scope, by outlining what is and isn't included in the system.
- Provides a clear record of the system's requirements which facilitates easier maintenance and future upgrades.
- Aligns all stakeholders, including clients, developers, and testers, by providing clear and accurate record of all requirements.

1.4 Preparation of the SRS

SRS is generally prepared during the initial stage of the Software Development Life Cycle (SDLC), before the design and implementation phase. This phase ensures that all stakeholders have clear understanding of the system's intended functionalities. It is prepared by Business Analysts, System Analysts, Project Managers, by working closely with the development team and key stakeholders to ensure that all requirements are

thoroughly understood and documented. This document is intended to direct the development, evaluation, and execution of the Fundora Crowdfunding Platform.

1.5 Intended Audience of the Document

The document is intended for:

- Project Stakeholders: To guarantee they have a comprehensive understanding of the system's functionalities and limitations.
- Development Team: To offer explicit instructions and comprehensive specifications for designing and creating the system.
- Quality Assurance Team: To define the standards and parameters for evaluating the system.
- End Users (Creators, Backers, Admins): To guarantee that their requirements and expectations are well understood and documented.
- Maintenance Team: To provide knowledge about the system's features for future upgrades and troubleshooting.
- Payment Gateway Partners: To understand integration requirements for eSewa and Khalti payment systems.

1.6 Definitions, Acronyms, and Abbreviations

Fundora: The crowdfunding platform name

SRS: Software Requirement Specifications

DFD: Data Flow Diagram

ERD: Entity-Relationship Diagram

API: Application Programming Interface

JWT: JSON Web Token

MERN: MongoDB, Express.js, React.js, Node.js

FN: Function

NF: Non-Functional Requirement

NPR: Nepalese Rupee

RAM: Random Access Memory

SSD: Solid-State Drive

GB: Gigabyte

Mbps: Megabits per second

IDE: Integrated Development Environment

CPU: Central Processing Unit

OS: Operating System

DBMS: Database Management System

GDPR: General Data Protection Regulation

ODM: Object Document Mapper

CI/CD: Continuous Integration/Continuous Deployment

SSL: Secure Sockets Layer

HTTPS: Hypertext Transfer Protocol Secure

2. System Overview

2.1 Project Description

Fundora is a next-generation crowdfunding platform designed specifically for Nepal that combines international crowdfunding best practices with localized payment solutions. The platform enables creators, innovators, and communities across Nepal to raise funds transparently and securely for their projects, creative works, and social causes.

2.2 Core Vision

To create a trust-driven ecosystem where creators can launch their ideas and backers can confidently support them — all through a transparent and verifiable digital environment tailored for the Nepali market.

2.3 Key Objectives

- Democratize Funding: Make crowdfunding accessible to all Nepali creators regardless of their location or background
- Build Trust: Implement transparent fund management through milestone-based releases and proof verification
- Local Integration: Seamlessly integrate with Nepal's digital payment ecosystem (eSewa, Khalti)
- Community Protection: Provide robust moderation and fraud detection mechanisms
- Direct Communication: Enable personal connection between backers and creators through real-time messaging

2.4 Target Users

- Creators: Individuals or organizations seeking funding for projects, creative works, or social causes
- Backers: Individuals wanting to support innovative projects and creative ideas
- Administrators: Platform moderators ensuring quality and preventing fraud

2.5 System Features Overview

The Fundora platform provides the following major features:

- Secure user authentication and profile management
- Comprehensive campaign creation and management tools
- Integrated payment processing through eSewa and Khalti
- Milestone-based fund release system with proof verification
- Community-driven flagging and administrative moderation
- Real-time one-to-one messaging between backers and creators
- Public commenting and campaign updates
- Role-based dashboards with analytics
- Multi-channel notification system
- Admin panel for platform management

3. Functional Requirements

3.1 Function 1: User Authentication and Management

The system provides secure access by providing options to register, login, and control accounts for creators, backers, and administrators.

Table 1 Functional Requirement User Authentication and Management

ID	Function	Description	Priority
FN 1.1	Register User	Users (creators, backers) can create accounts by providing necessary credentials including name, email, password, and role selection. Email verification is required.	High
FN 1.2	Login User	Registered users can securely log in using their credentials to access role-specific functionalities. System implements JWT-based authentication with refresh tokens.	High
FN 1.3	Password Management	Users can reset forgotten passwords via email link, change passwords from profile settings, and recover accounts securely.	High
FN 1.4	Profile Management	Users can update profile information including name, bio, profile picture, location, website, and social media links.	Medium
FN 1.5	Role-Based Access Control	System enforces role-based permissions where creators can create campaigns, backers can fund projects, and admins have	High

		moderation privileges.	
FN 1.6	Session Management	System automatically logs out users after 30 minutes of inactivity. Users can choose "Remember Me" option for extended sessions (7 days). Multiple simultaneous sessions are allowed with session tracking.	Medium

3.2 Function 2: Campaign Management

The system facilitates creating, viewing, editing, and managing crowdfunding campaigns with comprehensive details and media support.

Table 2 Functional Requirement Campaign Management

ID	Function	Description	Priority
FN 2.1	Create Campaign	Creators can create campaigns by providing title, description (minimum 100 characters), category, funding goal (minimum NPR 1,000), duration (7-90 days), funding type, and media uploads.	High
FN 2.2	Multi-Step Campaign Creation	Campaign creation follows a wizard approach with steps for: basic info, funding details, media upload, reward tiers, and milestone definition.	High
FN 2.3	Media Upload	Creators can upload up to 5 images and 1 video per campaign. Images are automatically optimized and stored on Cloudinary.	High

FN 2.4	Campaign Categories	System supports predefined categories: Technology, Art, Music, Film, Games, Education, Community, Innovation, Health, Environment.	Medium
FN 2.5	Reward Tier Management	Creators can define multiple reward tiers with title, description, amount, delivery date, quantity limits, and availability status.	High
FN 2.6	Milestone Definition	For milestone-based campaigns, creators define milestones with title, description, percentage, and order.	High
FN 2.7	Draft Saving	Campaigns can be saved as drafts and resumed later before submission for approval.	Medium
FN 2.8	Campaign Editing	Creators can edit pending or draft campaigns. Active campaigns with backers have restricted editing to prevent fraud	Medium
FN 2.9	Campaign Deletion	Creators can delete draft campaigns. Active campaigns cannot be deleted but can be cancelled with admin approval.	Low

3.3 Function 3: Campaign Discovery and Browsing

The system provides comprehensive search, filtering, and discovery features for backers to find campaigns of interest.

Table 3 Functional Requirement Campaign Discovery and Browsing

ID	Function	Description	Priority
FN 3.1	Browse Campaigns	Users can view all active campaigns in a grid layout with key information: title, image, funding progress, goal, backers, days remaining.	High
FN 3.2	Search Functionality	Users can search campaigns by title and description using text-based search with real-time results.	High
FN 3.3	Category Filtering	Users can filter campaigns by one or multiple categories to narrow down results.	High
FN 3.4	Sorting Options	Campaigns can be sorted by: Most Funded, Trending (most backed recently), Closing Soon, Recently Added.	High
FN 3.5	Pagination	Campaign listings support pagination with configurable items per page (default 12) to optimize load times.	Medium
FN 3.6	Campaign Detail View	Clicking a campaign shows full details including description, media gallery, funding progress, creator profile, rewards, milestones, and comments.	High
FN 3.7	Trending Campaigns	Homepage features trending campaigns based on recent backing activity and growth rate.	Medium

3.4 Function 4: Payment Integration

The system supports secure transactions through local payment gateways with multiple payment options and transaction tracking.

Table 4 Functional Requirement Payment Integration

ID	Function	Description	Priority
FN 4.1	eSewa Integration	System integrates eSewa payment gateway for secure NPR transactions. Users are redirected to eSewa portal for payment authentication.	High
FN 4.2	Khalti Integration	System integrates Khalti digital wallet for instant payments. Supports Khalti mobile app and web checkout.	High
FN 4.3	Payment Initialization	When backing a campaign, system creates transaction record with pending status and generates payment parameters for selected gateway.	High
FN 4.4	Payment Verification	System verifies payment completion through gateway API callbacks and updates transaction status accordingly.	High
FN 4.5	Payment Amount Validation	System enforces minimum backing amount (NPR 100) and validates reward tier amounts before payment initialization.	High
FN 4.6	Campaign Fund	Upon successful payment verification, campaign funded amount	High

	Update	and backer count are automatically updated in real-time.	
FN 4.7	Transaction History	Users can view complete transaction history with details: campaign name, amount, payment method, date, status, and receipt.	Medium
FN 4.8	Receipt Generation	System generates digital receipts for completed transactions with transaction ID and campaign details.	Medium
FN 4.9	Refund Processing	Admins can process refunds for failed campaigns or fraudulent activities with proper documentation.	Medium
FN 4.10	Payment Failure Handling	System handles payment failures gracefully, maintains transaction as failed, and allows users to retry.	High

3.5 Function 5: Milestone and Fund Release System

The system implements phased fund release based on milestone completion and proof verification to ensure project accountability.

Table 5 Functional Requirement Milestone and Fund Release System

ID	Function	Description	Priority
FN 5.1	Milestone Submission	After campaign reaches funding goal and ends, creators can submit proof for each milestone sequentially. Submission includes: selecting milestone from list, uploading proof files (images, videos, PDFs), detailed	High

		progress description, estimated completion date for next milestone. System validates submission is complete before allowing submission.	
FN 5.2	Proof Upload	Creators upload multiple proof documents: images, videos, documents. Files are uploaded to Cloudinary with progress indicators. System generates thumbnails for images and videos. Creators can caption each file. Files are accessible to admins for review and backers after approval.	High
FN 5.3	Admin Milestone Review	Admins access dedicated milestone review panel showing: pending submissions sorted by date, campaign details and funding amount, milestone number and description, creator's progress description, all uploaded proof files (viewable in lightbox), and action buttons (Approve, Reject, Request Resubmission). Admins can download all proof files as ZIP. Review guidelines available.	High
FN 5.4	Fund Release Calculation	Upon milestone approval, system calculates release amount: (total funded amount × milestone percentage) - platform fee (5%) - payment gateway fees (already	High

		deducted). Example: Campaign funded NPR 100,000, Milestone 1 is 25%: Release = (100,000 × 0.25) - 5,000 = NPR 20,000. System validates: sufficient unreleased funds available, calculation is accurate, and release doesn't exceed campaign total.	
FN 5.5	Fund Release Tracking	System maintains comprehensive fund release records: FundRelease document created with amount, milestone reference, release date, approval admin, payment method for disbursement, and status. Campaign's released_amount field updated. Timeline visualization shows all releases. Creators and backers can view release history. Financial reports include all releases for accounting.	High
FN 5.6	Creator Notifications	Creators receive notifications at key points: milestone proof submitted successfully, milestone under admin review (with average review time), milestone approved (with release amount and estimated transfer date), milestone rejected (with rejection reason and improvement suggestions), funds released to account (with transaction reference),	Medium

		and request for resubmission (with specific feedback).	
FN 5.7	Backer Progress Updates	Backers receive updates when: milestone proof is submitted (teaser without full details), milestone approved with summary (full proof accessible from campaign page), funds released to creator, and project completion. Updates include progress percentage and estimated project completion date. Backers can view proof files to verify progress.	Medium
FN 5.8	Milestone Status Tracking	Campaign detail page displays milestone timeline with visual progress: completed milestones (green checkmark), current milestone with proof submitted (yellow, under review), current milestone without proof (orange), future milestones (gray). Hovering shows details: submission date, approval date, proof preview. Color-coded status badges indicate: Pending, Submitted, Under Review, Approved, Rejected, Resubmission Required.	Medium
FN 5.9	Milestone Rejection	When admin rejects milestone: detailed reason required (dropdown + text), creator is notified immediately, funds remain locked for that milestone, creator can resubmit after	Medium

		making required changes, if all milestones rejected, case escalated to senior admin, and creators can appeal decision with additional evidence within 7 days.	
FN 5.10	Fund Disbursement	After approval, funds are transferred to creator's registered bank account or e-wallet: creator selects disbursement method during campaign setup (bank transfer, eSewa, Khalti), system initiates transfer through integrated payment APIs, creator receives confirmation email with transaction reference, transfer completed within 3-5 business days, and creators can track disbursement status in dashboard.	High

3.6 Function 6: Flagging and Moderation System

The system provides community-driven fraud detection and administrative moderation tools to maintain platform integrity.

Table 6 Functional Requirement Flagging and Moderation System

ID	Function	Description	Priority
FN 6.1	Flag Campaign	Backers and users can flag suspicious campaigns by clicking "Report" button, selecting reason from dropdown (Fraud/Scam,	High

		Misleading Information, Inappropriate Content, Copyright Violation, No Progress Updates, Spam, Other), providing detailed description (required, min 100 characters), optionally uploading evidence screenshots (up to 3 images, 5MB each), and submitting. User receives confirmation and flag ID for tracking.	
FN 6.2	Duplicate Flag Prevention	System prevents spam flagging: checks if user already flagged this campaign with status pending/under_review, shows warning "You've already reported this campaign", allows viewing previous flag status and adding comments to existing flag, but prevents creating duplicate flag. Users can flag different campaigns freely. Database has unique constraint on (campaign_id, user_id, active_flag).	Medium
FN 6.3	Flag Counter	System tracks active flag count per campaign: displayed on admin panel with color coding (0-2 green, 3-4 yellow, 5+ red), sorted by flag count for priority review, visible in campaign admin details, used for auto-suspension trigger, and historical flag data retained for analysis. Flag metrics help identify problem	Medium

		patterns.	
FN 6.4	Admin Flag Review	Admins access comprehensive flag management panel: list all flags filtered by status (pending, under_review, resolved, dismissed), sort by campaign, date, flag count, view campaign full details with statistics (backers, amount, age), see all flags for campaign with reporter information, read each flag's reason and evidence, and view campaign creator's history. Panel includes bulk actions for efficiency.	High
FN 6.5	Flag Resolution	Admins can take multiple actions: Uphold Flag (if valid) with action options: warn creator, request corrections, suspend for 7 days, terminate campaign, or Dismiss Flag (if invalid/spam) with reasons: insufficient evidence, campaign is legitimate, misunderstanding. All resolutions require admin comments explaining decision. Both reporter and creator are notified of outcome with reasoning.	High
FN 6.6	Campaign Termination	When admin terminates campaign for verified fraud/violation: campaign status changes to "terminated", immediate removal from all public areas, automatic refund initiated for	High

		all backers (refund process begins within 24 hours), creator account suspended pending investigation, termination reason posted publicly, all transactions marked for refund, and case documented for legal records. Terminated campaigns remain in admin area for audit.	
FN 6.7	Flag Notifications	Flag reporters receive updates: flag received and under review (immediate), flag assigned to admin (within 24 hours), admin decision made (uphold or dismiss) with brief explanation, and action taken if upheld. Reporters can appeal dismissed flags with additional evidence within 7 days. Email notifications for all updates.	Low
FN 6.8	Creator Warning System	Progressive discipline for flagged creators: First Issue - warning email + mandatory violation explanation, Second Issue - 7-day suspension + required compliance training, Third Issue - 30-day suspension + account review, Fourth Issue - permanent ban from creating campaigns. Creators can view warning history and current standing in dashboard.	Medium
FN 6.9	False Flag Protection	To prevent abuse of flagging system: users flagging legitimate campaigns	Low

		repeatedly (3+ false flags) face penalties, admins can mark flags as "malicious", malicious flaggers have flagging privilege restricted for 30 days, and persistent abuse results in account suspension. System tracks flag accuracy per user.	

3.7 Function 7: Comment and Interaction System

The system enables public communication between creators and backers through comments, updates, and community engagement.

Table 7 Functional Requirement Comment and Interaction System

ID	Function	Description	Priority
FN 7.1	Post Comments	Authenticated users can post comments on campaign pages to ask questions or provide feedback. Comments limited to 1000 characters. Rich text editor supports: bold, italic, bullet points, and hyperlinks. Comments display user's name, profile picture, timestamp, and "Backer" or "Creator" badge if applicable. Comments appear in chronological order (newest first/oldest first toggle).	Medium
FN 7.2	Reply to Comments	Users can reply to any comment creating threaded conversations. Replies indented visually and show	Medium

		parent comment context. Maximum nesting depth of 3 levels prevents excessive threading. Reply notifications sent to original commenter. Thread can be collapsed/expanded for readability.	
FN 7.3	Edit Comments	Users can edit their own comments within 24 hours of posting. Edited comments show "Edited" tag with timestamp. Edit history not publicly visible but logged in database for moderation. Prevents abuse while allowing correction of mistakes.	Low
FN 7.4	Delete Comments	Users can delete their own comments anytime. Deleted comments show "[Comment deleted by user]" placeholder if replies exist (preserves thread structure). Admins can delete any comment violating platform policies with reason logged. Comment deletion notifications sent to reply authors.	Medium
FN 7.5	Campaign Updates	Creators can post updates about project progress visible to all users (logged in and guests). Updates support rich text, images (up to 3, 5MB each), and videos (up to 100MB). Updates appear in chronological feed on campaign page with separate "Updates" tab. Backers	Medium

		receive email notification for each update. Updates cannot be deleted, only new ones added.	

3.8 Function 8: Real-Time Messaging System

The system provides secure, private one-to-one real-time communication between backers and campaign creators for direct inquiries and personalized support.

Table 8 Functional Requirement Real-Time Messaging System

ID	Function	Description	Priority
FN 8.1	Initiate Conversation	Only backers who have financially contributed to a campaign can initiate private conversations with campaign creators. "Message Creator" button appears on campaign page after successful backing. Button shows creator's typical response time (calculated from past conversations). Clicking opens conversation window or creates new conversation if first time. Prevents spam from non-backers.	High
FN 8.2	Real-Time Messaging	Users send and receive text messages instantly using WebSocket/Socket.io technology. Messages deliver in real-time when both parties online. Offline messages queued and delivered when recipient comes online. Character limit of 2000	High

		per message. Messages support line breaks and basic formatting. No lag or refresh needed - appears immediately in conversation.	
FN 8.3	Message History	All conversations persistently stored in MongoDB. Users can scroll infinitely through message history (paginated in database, loads 50 messages at a time). Messages show timestamp (relative for recent: "2 minutes ago", absolute for old: "Nov 15, 2:30 PM"). Search within conversation by keyword. Export conversation history as PDF for records. History preserved even if campaign ends.	High
FN 8.4	Message Notifications	Users receive multi-channel notifications for new messages: In-app notification badge on message icon (unread count), Browser push notification with message preview (if permitted by user), Email notification (if user offline > 1 hour, configurable), and Sound alert (customizable ding sound). Notifications stack (multiple messages show count). Preview shows sender name and first 50 characters of message.	High
FN 8.5	Conversation List	Dedicated inbox page shows all conversations in left sidebar sorted	High

		by most recent activity (last message time). Each conversation preview shows: Contact name with profile picture, Campaign name (small text), Last message preview (truncated to 60 characters), Timestamp of last message, Unread count badge if applicable, Online status indicator. Empty state for new users: "No conversations yet. Back a campaign to start messaging!"	
FN 8.6	Block/Report User	Users can block abusive contacts from conversation options menu. Blocking prevents: Blocked user sending new messages, Blocked user seeing online status, Notifications from blocked user. Existing conversation hidden from list but data retained. Users can report inappropriate messages to admins with reason (Spam, Harassment, Inappropriate Content, Fraud). Admins review reports in moderation panel. Persistent offenders face account suspension.	Medium

3.9 Function 9: Dashboard and Analytics

The system provides role-specific dashboards with relevant information, statistics, and management tools.

Table 9 Functional Requirement Dashboard and Analytics

ID	Function	Description	Priority
FN 9.1	Backer Dashboard	Backers have personalized dashboard showing: Active campaigns backed (with progress bars and days remaining), Completed campaigns backed (with project status), Total amount backed (sum across all campaigns), Number of campaigns backed, Recent transactions (last 10 with links to receipts), Active conversations with creators (unread count highlighted), Campaign updates from backed projects, Recommended campaigns based on backing history. Quick action buttons: View All Transactions, Message Creators, Browse More Campaigns.	High
FN 9.2	Creator Dashboard	Creators see comprehensive dashboard: Campaign overview cards (each campaign with status, funding, backers), Total funds raised (across all campaigns), Active backers (unique backer count), Pending milestone reviews (alert badge), Recent backing activity (last 10 backers with amounts), Message center (conversations needing response highlighted), Campaign	High

		performance chart (funding over time), Conversion rate (views to backers ratio), Quick actions: Create New Campaign, Submit Milestone Proof, Reply to Messages, Post Update.	
FN 9.3	Admin Dashboard	Admins access platform-wide statistics: Total campaigns (with status breakdown pie chart), Total users (creators vs backers), Total funding amount (NPR with trend), Active campaigns (currently live count), Pending campaign approvals (requires action count), Pending milestone reviews (requires action count), Flagged campaigns (requires review count), Platform success rate (% of campaigns reaching goal), Monthly revenue (platform fees collected), Recent activity feed (new campaigns, large backing, flags). Quick navigation to: Approve Campaigns, Review Milestones, Review Flags, User Management.	High
FN 9.4	Campaign Analytics	Creators view detailed analytics for each campaign: Funding trend line chart (daily funding amounts over campaign duration), Backer demographics (location if provided, average pledge amount), Traffic	Medium

		sources (direct, social media, search, referrals), Conversion funnel (views → detailed views → backing → completed payment), Daily/Weekly/Monthly view options, Comparison with similar campaigns (category average), Peak backing times (day of week, time of day heatmap), Reward tier popularity (which tiers most selected), Comment engagement metrics. Export all data as CSV.	
FN 9.5	Platform Analytics	Admins see comprehensive platform analytics: Funding trends (total funding per month line chart over past year), Category performance (which categories have most campaigns, highest success rate, most funding), User growth (new users per month, retention rate), Success rates by campaign type (reward vs donation vs milestone), Average funding amount per campaign, Average number of backers per campaign, Platform fee revenue tracking, Payment method distribution (eSewa vs Khalti usage), Geographic distribution of users (by city if location provided). Interactive filters: date range, category,	Medium

		campaign status.	
FN 9.6	Export Reports	All dashboard sections have export functionality: Transaction history (CSV with all fields: date, campaign, amount, status, payment method), Campaign analytics (CSV with daily stats or PDF summary report), Backer list (for creators: CSV with backer names, amounts, rewards, contact info with permission), Financial reports (PDF for tax purposes with official Fundora stamp), Platform analytics (admins only: comprehensive Excel workbook with multiple sheets). Export respects user privacy (only data user has access to).	Low
FN 9.7	Message Center	Dedicated messaging section in all dashboards: Inbox showing all conversations (sorted by unread, then recent), Unread count badge prominently displayed, Quick reply without leaving dashboard, Filter conversations by campaign, Mark all as read button, Search across all conversations, Archive/Delete bulk actions. Integration with main dashboard (recent messages widget). Notification bell for new messages.	High

FN 9.8	Activity Feed	Real-time activity feed shows relevant events: For Backers - campaigns you backed received new backer, campaign you backed posted update, milestone approved for backed campaign, creator responded to your message. For Creators - new backer, new message, campaign approved/rejected, milestone reviewed, campaign trending. For Admins - new campaign submitted, new flag received, unusual activity detected. Feed updates without refresh. Click item to go to relevant page.	Medium

3.10 Function 10: Admin Panel and Management

The system provides comprehensive administrative tools for platform management, moderation, and quality control.

Table 10 Functional Requirement Admin Panel and Management

ID	Function	Description	Priority
FN 10.1	Campaign Approval	Admins review pending campaigns in dedicated approval queue: List shows campaign title, creator name, category, goal amount, submission date, preview thumbnail. Click to view full campaign details including description, images, video, rewards, milestones. Approval checklist:	High

		follows community guidelines, realistic goal and rewards, clear description, appropriate media. Actions: Approve (campaign goes live immediately, creator notified), Reject with reason (dropdown + custom message: unrealistic goals, inappropriate content, incomplete information, copyright issues, etc.). Bulk approve option for clearly acceptable campaigns. Average review time tracked (goal < 24 hours).	
FN 10.2	User Management	Admins can manage all platform users: User list with search by name/email/ID, filters (role: backer/creator/admin, status: active/suspended/deactivated, date joined), User detail page shows: profile info, registration date, campaigns created/backed, total funding given/received, transaction history, message history (for violations), warning/suspension history. Actions: View full profile, Suspend account (with reason and duration: 7/14/30/90 days or permanent), Unsuspend account, Reset password (sends email), Delete account (after 90-day grace period), Upgrade role (backer → creator or →	Medium

		admin). Suspension reasons logged and emailed to user with appeal instructions.	
FN 10.3	Transaction Monitoring	Admins monitor all platform transactions: Transaction list with comprehensive filters (status: all/completed/pending/failed/refunded, date range picker, amount range, payment method: eSewa/Khalti, campaign search, user search), Export filtered transactions as Excel for accounting. Transaction details page: full transaction info, associated campaign and users, payment gateway reference, timeline of status changes, refund option if eligible. Anomaly detection highlights: large transactions (> NPR 50,000), rapid multiple transactions, failed payment patterns. Total transaction volume and value widgets. Daily/weekly/monthly reconciliation reports.	Medium
FN 10.4	Content Moderation	Admins have content moderation tools: Review flagged comments (reason, context, reporter, date), View flagged messages (from user reports, limited preview for privacy), Review reported campaigns (flags from FN 6.1), Action options: Delete content	Medium

		(with notification), Warn user (email warning + dashboard notice), Suspend user (temporary or permanent), Dismiss report (if false alarm). Moderation logs all actions with admin name, timestamp, reason. Bulk moderation for spam. Automated filters flag potential violations: profanity, scam keywords, suspicious URLs. Weekly moderation summary report.	
FN 10.5	Platform Settings	Admins configure global platform settings (superadmin only): Financial settings (platform fee percentage: default 5%, minimum pledge amount: default NPR 100, maximum campaign goal: default NPR 10,000,000), Campaign settings (min/max duration: 7-90 days, approval required toggle, auto-suspend flag threshold: default 5), Payment gateways (enable/disable eSewa/Khalti, configure merchant IDs, test mode toggle), Email settings (SMTP config, notification templates editing), Feature toggles (enable/disable messaging, commenting, etc.), Maintenance mode (disable new campaigns/backing for updates). All changes logged. Settings validation	Low

		before saving.	
FN 10.6	System Logs	Admins access system logs for debugging and security: Error logs (server errors with stack traces, severity levels), Security logs (failed login attempts, suspicious activities, IP addresses), Audit logs (all admin actions with details), Payment logs (gateway communications, verifications), Email logs (emails sent, delivery status), User activity logs (campaign views, searches, high-level actions only for privacy). Logs filterable by: severity, date range, user, action type. Search by keyword. Export logs for external analysis. Logs retained 90 days (critical logs: 1 year).	Low
FN 10.7	Message Monitoring	Admins can review reported messages in moderation panel: List of reported messages with: reporter name and reason, conversation context (5 messages before/after for context), timestamp and participants, reason for report, status (pending/reviewed/dismissed). Admin actions: View full conversation (limited to reported thread only for privacy), Warn sender, Suspend sender account, Delete message, Dismiss report (if false), Block communication	Medium

		between parties. Admins cannot read unreported messages (privacy protected). Sensitive reports (harassment, threats) escalated to superadmin.	
FN 10.8	Analytics Dashboard	Admins view platform health metrics: Real-time stats (active users now, campaigns live, pending approvals count), Growth charts (new users per day, campaigns per day, funding per day), Engagement metrics (average time on site, pages per session, bounce rate), Conversion funnels (visitor → registered → backer → creator), Success metrics (campaign success rate trend, average funding percentage, average backer count), Revenue tracking (platform fees collected, projected monthly revenue). Custom date ranges and comparison periods (compare this month vs last month). Downloadable reports for stakeholders.	Medium

3.11 Function 11: Notification System

The system keeps users informed through real-time notifications and email alerts for important events.

Table 11 Functional Requirement Notification System

ID	Function	Description	Priority
FN 11.1	In-App Notifications	Users receive in-app notifications (bell icon in header) for important events: Campaign Events - backed campaign has new update, backed campaign milestone approved, backed campaign about to end (48 hours), backed campaign succeeded/failed. Interaction Events - someone commented on your comment, creator responded to your comment, someone liked your comment. Messaging Events - new message received (separate from message badge). Account Events - campaign approved/rejected, milestone review completed, account warning/suspension. Payment Events - payment successful, payment failed, refund processed. Admin Events (admins only) - new campaign pending approval, new flag received, milestone pending review. Notifications appear in dropdown (max 50 most recent, load more option). Unread count badge on bell icon. Mark as read individually or all at once. Clicking notification navigates to relevant page.	High

FN 11.2	Notification Center	Dedicated notifications page shows all notifications: Tabs: All, Unread, Read, filter by type dropdown (campaigns, messages, payments, account), date range filter, search notifications by keyword. Each notification shows: icon (color-coded by type), title, description, timestamp (relative: "2 hours ago"), read/unread indicator, action button ("View Campaign", "Reply", "Review", etc.). Pagination (50 per page). Bulk actions: mark all as read, clear read notifications (older than 30 days). Notification preferences link. Empty state for new users with explanation.	Medium
FN 11.3	Email Notifications	System sends email notifications for critical events users might miss in-app: Welcome email upon registration (verify email link, getting started guide), campaign approval/rejection (immediate), backing confirmation (immediate with receipt attached as PDF), payment success/failure (immediate), new message (if user offline > 1 hour, batched to prevent spam), campaign update from backed project (within 1 hour, batched if multiple), milestone approved for backed campaign	Medium

		(immediate), funds released to creator (immediate with transaction details), campaign ending soon (48 hours before end), account warnings/suspensions (immediate). All emails branded (Fundora logo, colors) with unsubscribe link. HTML + plain text versions. User can customize email preferences.	

4. External Interface Requirements

4.1 Hardware Requirements During Development

The following hardware configurations are assumed during the development stage of Fundora to facilitate an efficient implementation process:

- Development Servers: Minimum 16 GB RAM, 512 GB SSD storage, and quad-core processor (Intel i5/i7 or AMD Ryzen 5/7) for running MongoDB database, Node.js development server, Redis (if used for caching), and development tools simultaneously without performance degradation.
- Network Infrastructure: Stable high-speed internet connection of at least 50 Mbps (100 Mbps recommended) for seamless API testing with external services (eSewa, Khalti sandbox environments), real-time Cloudinary media uploads (images/videos), Git operations (code push/pull), npm package downloads, and WebSocket connection testing for messaging feature.
- Developer Workstations: High-capacity systems with at least 8 GB RAM (16 GB recommended), 256 GB SSD storage (512 GB recommended), and multi-core processor (quad-core minimum) for running modern IDEs (VS Code, WebStorm), multiple browser instances with developer tools, Docker containers, database management tools (MongoDB Compass), API testing tools (Postman), and local development servers.

- Testing Devices: Multiple physical and virtual devices for cross-platform testing including: Desktop computers (Windows 10/11, macOS 11+), tablets (iOS iPad, Android tablets), smartphones (iOS 12+, Android 8+), various screen sizes (1366x768 to 4K resolution). Ensures responsive design works across all target devices.
- Backup Systems: External hard drives or network-attached storage (NAS) with minimum 1 TB capacity for regular backups of: source code repositories, database dumps (MongoDB exports), media assets (images/videos for testing), development documentation, and sensitive configuration files. Automated daily backups recommended.

4.2 Software Requirements During Development

The following software and tools are required for developing the Fundora platform:

- Database Management System:
    - MongoDB 6.x or higher (NoSQL database) for development and testing
    - MongoDB Atlas free tier (M0 Sandbox, 512 MB storage) for cloud database testing
    - MongoDB Compass (GUI) for database visualization, query building, and data management
    - Robo 3T/Studio 3T (alternative GUI tools)
- Operating System:
    - Windows 10/11 Professional (64-bit)
    - macOS 11 Big Sur or later (for Mac developers)
    - Linux distributions: Ubuntu 20.04 LTS+, CentOS 8+, Fedora 34+ (compatible with Node.js)
    - WSL2 (Windows Subsystem for Linux) for Windows developers preferring Linux environment
- Development Environment:
    - Node.js: v18.x LTS (Long Term Support) with npm (v9+) or Yarn (v1.22+) package manager or above

    - IDEs: Visual Studio Code (recommended, free) with extensions (ESLint, Prettier, MongoDB for VS Code), WebStorm/IntelliJ IDEA (paid, optional), or Atom/Sublime Text
    - Version Control: Git 2.30+ for source control, GitHub Desktop (GUI option for Git), Git Bash/Terminal for command-line operations
    - API Testing: Postman (latest version) for API endpoint testing, request/response validation, environment management
    - Containerization: Docker Desktop (latest) for containerizing application, consistent development environments, and microservices testing
- Backend Development:
    - Framework: Express.js 4.x (Node.js web framework)
    - ODM: Mongoose 7.x (MongoDB Object Document Mapper)
    - Libraries:
        - jsonwebtoken (JWT authentication)
        - bcryptjs (password hashing)
        - express-validator (input validation)
        - multer (file upload handling)
        - cookie-parser (cookie management)
        - cors (Cross-Origin Resource Sharing)
        - helmet (security headers)
        - express-rate-limit (rate limiting)
        - winston/morgan (logging)
        - node-cron (scheduled tasks)
        - nodemailer (email sending)
        - socket.io (WebSocket for real-time messaging)
        - axios (HTTP client for payment gateway APIs)
- Frontend Development:
    - Framework: React 18.x (UI library)
    - Build Tool: Vite 4.x (fast build tool and dev server)
    - State Management: Redux Toolkit 1.9+ (global state management)
    - Routing: React Router v6 (client-side routing)

    - Styling: Tailwind CSS 3.x (utility-first CSS framework)
    - HTTP Client: Axios (API communication)
    - Server State: React Query/TanStack Query (server state management)
    - Forms: Formik 2.x + Yup (form handling and validation)
    - Icons: Lucide React (icon library)
    - Charts: Recharts (data visualization)
    - Notifications: React-Toastify (toast notifications)
- Cloud Services:
    - Cloudinary: Account (free tier: 25 GB storage, 25 GB bandwidth/month) for media storage, image optimization, and CDN delivery
    - MongoDB Atlas: Cluster (free tier: M0 Sandbox, 512 MB) for managed database hosting with automated backups
    - Email Service: SendGrid/Amazon SES free tier (optional, for transactional emails)
- Payment Gateway Integration:
    - eSewa: Merchant Test/Sandbox Account for development and testing
    - Khalti: Developer API credentials (test environment) for integration testing
    - Sandbox documentation and test cards/accounts from both providers
- Testing Tools:
    - Unit Testing: Jest (JavaScript testing framework) with @testing-library/react
    - API Testing: Supertest (HTTP assertion library)
    - E2E Testing: Cypress (end-to-end testing framework) with Chrome browser
    - Code Quality: ESLint (JavaScript linting) + Prettier (code formatting), Husky (Git hooks for pre-commit checks)
- Additional Tools:
    - Browser: Chrome/Firefox/Safari (latest versions) with React Developer Tools and Redux DevTools extensions
    - Design Tools: Figma/Adobe XD (optional, for viewing UI designs)

4.3 Hardware Requirements During Usage

End users will need the following hardware configurations for optimal usage of the Fundora platform:

- Desktop/Laptop Computers:
    - RAM: Minimum 4 GB (8 GB recommended for smooth multitasking)
    - Storage: 128 GB available (for browser cache, downloaded receipts, and operating system)
    - Processor: Dual-core CPU minimum (Intel Core i3 or AMD equivalent), quad-core recommended for better performance
    - Display: Minimum resolution 1366x768 (HD), recommended 1920x1080 (Full HD) or higher for optimal viewing of campaign images and analytics
    - Internet: Stable connection minimum 5 Mbps (10+ Mbps recommended for smooth video viewing, image loading, and real-time messaging)
    - Input Devices: Keyboard and mouse/trackpad for navigation and form input
- Mobile Devices (Smartphones):
    - RAM: Minimum 2 GB (3-4 GB recommended for seamless experience)
    - OS: Android 8.0 Oreo or higher, iOS 12 or higher (for modern browser support and security)
    - Display: Minimum 5-inch screen (larger screens 6+ inches provide better experience for browsing campaigns)
    - Storage: Minimum 2 GB free space (for browser cache and app data if PWA installed)
    - Camera: Functional camera for creators uploading campaign photos/proof (8 MP+ recommended)
    - Internet: 3G minimum, 4G/LTE or WiFi recommended for media uploads and real-time messaging

- Tablets:
    - Display: 7-inch minimum (10+ inches recommended for comfortable browsing and campaign management)
    - RAM: Minimum 2 GB (4 GB recommended)
    - OS: Android 8.0+, iOS 12+, or iPadOS
    - Storage: 32 GB minimum
    - Internet: Stable WiFi or mobile data connection
- Network Requirements:
    - Minimum Bandwidth: 5 Mbps for basic browsing and backing campaigns
    - Recommended Bandwidth: 10+ Mbps for uploading campaign images/videos, viewing video campaigns, real-time messaging with file sharing
    - Latency: < 100ms recommended for real-time features (typing indicators, online status updates, instant message delivery)
    - Stability: Consistent connection preferred; platform gracefully handles temporary disconnections with message queuing and auto-reconnect

4.4 Software Requirements During Usage

End users require the following software for accessing and using the Fundora platform:

- Operating Systems:
    - Desktop: Windows 10/11, macOS 10.14 Mojave or higher, Linux distributions (Ubuntu 18.04+, Fedora, etc.)
    - Mobile: Android 8.0 Oreo or higher, iOS 12 or higher
    - Tablet: iPadOS 12+, Android 8.0+
- Web Browsers (Must be latest versions or updated within last 2 years):
    - Recommended: Google Chrome 90+ (best performance, full feature support)
    - Supported: Mozilla Firefox 88+, Safari 14+ (macOS/iOS), Microsoft Edge 90+ (Chromium-based)

    - Mobile: Chrome Mobile 90+, Safari Mobile 14+ (iOS), Samsung Internet 14+
    - Required Browser Features: JavaScript enabled (mandatory), Cookies enabled (for authentication), LocalStorage support (for session management), WebSocket support (for real-time messaging), HTML5 support (for media playback), SSL/TLS 1.2+ (for secure connections)
- Browser Settings and Requirements:
    - JavaScript: Must be enabled (platform will not function without it)
    - Cookies: First-party cookies required for authentication (refresh tokens)
    - Pop-ups: Allow pop-ups from Fundora domain for payment gateway redirects
    - Screen Resolution: Minimum 375px width (iPhone SE), recommended 1366x768 or higher
    - Color Depth: 24-bit color minimum
    - Responsive Design: Platform adapts to screen sizes from 375px (mobile) to 4K displays
- Payment Applications (Required for backing campaigns):
    - eSewa:
        - eSewa mobile app (latest version) installed on smartphone for mobile payments
        - OR active eSewa web account accessible through browser
        - Verified eSewa account with sufficient balance or linked bank account
    - Khalti:
        - Khalti mobile app (latest version) for digital wallet payments
        - OR Khalti web account (browser-based)
        - Verified Khalti account with KYC completed
- Security Software and Best Practices:

        - Antivirus: Updated antivirus software recommended (not mandatory but advised for security)
        - Firewall: Windows Firewall or equivalent enabled
        - HTTPS: Platform enforces HTTPS; users should verify secure connection (padlock icon in browser)
        - Password Manager: Recommended for secure password management (optional)
        - Two-Factor Authentication: Supported for enhanced account security (optional but recommended)
- Additional Software (Optional but Enhanced Experience):
    - PDF Reader: Adobe Acrobat Reader, browser built-in PDF viewer, or equivalent for viewing receipts and reports
    - Email Client: Any email application (Gmail app, Outlook, Apple Mail) for receiving notifications and password resets
    - Image Viewer: Default system image viewer for downloaded campaign images
    - Video Player: Modern browsers have built-in HTML5 video players; no additional software needed

5. Non-Functional Requirements

Non-Functional requirements define the quality attributes, system performance, and constraints that must be met by the software for it to provide a seamless and secure user experience.

5.1 Non-Functional Requirement 1: Performance

Table 12 Non-Functional Requirement Performance

ID	Category	Description
NF 1	Performance	1. All application pages should load within 3 seconds on standard broadband connection (10 Mbps) to ensure smooth user experience and prevent user

		drop-off. 2. API responses should return within 500ms for simple queries (single record lookup) and maximum 2 seconds for complex database operations (joins, aggregations, full-text search) 3. Image loading should be optimized with lazy loading (images load as user scrolls), progressive rendering (low-quality placeholder first, then full quality), and WebP format conversion where supported by browser. 4. Campaign search and filtering should return results within 2 seconds even with multiple active filters and search keywords, using database indexes for optimization. 5. Payment gateway redirects should complete within 5 seconds, with progress indicator showing status to user during process. 6. Dashboard analytics and charts should load within 4 seconds with skeletal loading indicators showing data structure while loading. 7. Real-time messaging should deliver messages with latency under 1 second when both users online, using WebSocket for instant delivery. 8. File uploads (images, videos) should show accurate progress percentage and estimated time remaining, with chunked upload for files over 10MB.

5.2 Non-Functional Requirement 2: Scalability

Table 13 Non-Functional Requirement Scalability'

ID	Category	Description
NF 2	Scalability	1. The system should handle at least 1,000 concurrent users (simultaneous active sessions) without performance degradation, with ability to scale to 5,000+ users during peak times (campaign launches, promotional events). 2. Database should support growth to 10,000+ active campaigns, 50,000+ registered users, and 100,000+ transactions within first year of operation without requiring architecture changes. 3. Cloud infrastructure (MongoDB Atlas for database, Cloudinary for media) should auto-scale based on demand, with automatic resource allocation during traffic spikes. 4. API should handle 100+ requests per second during normal operation and burst capacity of 500+ requests per second during peak usage, with load balancing if needed. 5. File upload system should support batch uploads and concurrent processing, handling 50+ simultaneous file uploads across platform. 6. Real-time messaging system should support 500+ concurrent active chat sessions with WebSocket connections, with connection pooling and efficient message queuing. 7. System architecture should support horizontal

		scaling by adding more server instances (stateless design, session storage in database/Redis, no server affinity requirements). 8. Database queries should be optimized with proper indexing, query plan analysis, and caching strategies to maintain performance as data grows.

5.3 Non-Functional Requirement 3: Responsiveness

Table 14 Non-Functional Requirement Responsiveness

ID	Category	Description
NF 3	Responsiveness	1. User interface should be fully responsive across all device categories: desktop (1920px+ resolution), laptop (1366px-1920px), tablet (768px-1024px), and mobile (375px-768px) with fluid layouts and adaptive components. 2. Search results should appear within 2 seconds with auto-suggestion appearing after user types 3 characters, providing instant feedback and reducing unnecessary searches. 3. Form validations should provide instant feedback without page reload, showing error messages immediately below relevant fields as user completes input (on blur or keystroke). 4. Real-time updates for funding progress should reflect within 5 seconds of payment completion, using WebSocket push notifications to update all viewers of campaign page simultaneously. 5. Notification badges (unread count) should update in

		real-time when new notifications or messages arrive, without requiring page refresh or manual action. 6. Touch targets on mobile devices should be minimum 44x44 pixels (iOS guideline) for easy tapping with finger, with adequate spacing between adjacent clickable elements. 7. Page transitions and animations should be smooth (60fps target) with hardware acceleration where possible, using CSS transforms and requestAnimationFrame for JavaScript animations. 8. Typing indicators in messaging should appear within 500ms of user starting to type, updating in real-time as other party composes message.

5.4 Non-Functional Requirement 4: Security

Table 15 Non-Functional Requirement Security

ID	Category	Description
NF 4	Security	1. All sensitive data (user passwords, payment information, personal details) must be encrypted using industry-standard algorithms: bcrypt (10 salt rounds minimum) for password hashing, AES-256 for stored sensitive data at rest. 2. All API communications and web traffic must use HTTPS/SSL encryption (TLS 1.2 or higher) with valid SSL certificate, enforcing secure connections and redirecting HTTP to HTTPS automatically. 3. JWT tokens for authentication should expire after

		15 minutes (access token) with secure refresh mechanism using HTTP-only cookies (refresh token, 7-day expiry), preventing XSS attacks on tokens. 4. Payment processing must comply with PCI DSS standards through certified payment gateways (eSewa, Khalti), with Fundora never storing or processing raw credit card data, only transaction references. 5. SQL/NoSQL injection prevention through parameterized queries (Mongoose ODM), input sanitization (express-mongo-sanitize), and validation of all user inputs before database operations. 6. Cross-Site Scripting (XSS) protection through Content Security Policy headers, input encoding/escaping, and sanitization of user-generated content (campaign descriptions, comments) before display. 7. Cross-Site Request Forgery (CSRF) protection for all state-changing operations (POST, PUT, DELETE) using CSRF tokens, with validation on server side. 8. Rate limiting to prevent brute force attacks: 5 login attempts per 15 minutes per IP address, 100 API requests per minute per user, with temporary IP blocking for persistent abuse. 9. Session management with automatic logout after 30 minutes of inactivity, detection of suspicious login patterns (new device, unusual location), and optional two-factor authentication for sensitive operations.

		10. Regular security audits, penetration testing before production deployment, dependency vulnerability scanning (npm audit), and prompt patching of security vulnerabilities. 11. File upload security: file type validation (whitelist allowed extensions), file size limits enforced, virus/malware scanning before storage, and generated filenames (no user-supplied filenames stored). 12. Database security: connection strings in environment variables, database user permissions (least privilege principle), and regular backup encryption.

5.5 Non-Functional Requirement 5: Usability

Table 16 Non-Functional Requirement Usability

ID	Category	Description
NF 5	Usability	1. User interface should follow consistent design patterns (colors, typography, spacing, button styles) with intuitive navigation throughout application, adhering to established UI/UX best practices and maintaining visual hierarchy. 2. New users should be able to create account, browse campaigns, and understand how to back a campaign within 5 minutes without needing external tutorials or help documentation, with clear call-to-action buttons and guided workflows. 3. Campaign creation wizard should guide creators step-by-step with clear instructions (numbered steps

		1-5), progress indicators showing completion percentage, helpful tooltips on hover for complex fields, and example campaigns for reference. 4. Error messages should be clear, specific, and provide actionable solutions (e.g., "Password must be at least 8 characters" instead of "Invalid password"), positioned near relevant form field with red color coding and error icon. 5. System should support both English and Nepali languages for better local accessibility, with easy language switcher in header, right-to-left text support where applicable, and culturally appropriate icons and imagery. 6. Help documentation and FAQs should be easily accessible from all pages via help icon in navigation, searchable knowledge base, step-by-step guides with screenshots, and video tutorials for complex processes. 7. Color contrast should meet WCAG 2.1 Level AA standards (minimum 4.5:1 for normal text, 3:1 for large text) for accessibility, ensuring readability for users with visual impairments and in bright/dim lighting conditions. 8. Form fields should have clear labels positioned above inputs, placeholder text providing example formats, asterisks indicating required fields, and validation feedback with helpful messages explaining requirements.

		9. Loading states should be visually indicated with spinners or progress bars (not blank screens), skeletal loaders showing content structure, and estimated time remaining for long operations like file uploads. 10. Success confirmations should be clearly communicated with visual feedback (green checkmark icons), success messages in green banners, and confirmation emails for important actions (payment, campaign approval).

5.6 Non-Functional Requirement 6: Reliability

Table 17 Non-Functional Requirement Reliability

ID	Category	Description
NF 6	Reliability	1. System uptime should be 99.5% or higher (allowing maximum 3.65 hours downtime per month), measured over monthly periods, excluding scheduled maintenance windows announced in advance. 2. Automated database backups should run daily at 3:00 AM NPT with incremental backups every 6 hours, 30-day retention for daily backups, 90-day retention for weekly backups, and 1-year retention for monthly backups. 3. Payment transactions should have 99.9% success rate (excluding user errors like insufficient funds) with proper error handling, automatic retry mechanisms for network failures (max 3 retries with exponential backoff), and immediate user notification of transaction status.

		4. System should gracefully handle failures with appropriate error messages (not exposing technical details to users), fallback mechanisms (cached data when API fails), and automatic recovery from transient errors without manual intervention. 5. Database transactions should be ACID-compliant (Atomicity, Consistency, Isolation, Durability) ensuring data consistency even during concurrent operations, with transaction rollback on failures to prevent partial data updates. 6. Failed payment attempts should not result in duplicate charges (idempotency keys) or lost data, with transaction logging for audit trail and reconciliation, and proper refund workflows for any accidental charges. 7. System should automatically recover from non-critical errors (e.g., email sending failure) with retry queues, dead letter queues for persistent failures, and alerting system for operations team on critical failures.

5.7 Non-Functional Requirement 7: Maintainability

Table 18 Non-Functional Requirement Maintainability

ID	Category	Description
NF 7	Maintainability	1. Code should follow consistent naming conventions (camelCase for variables/functions, PascalCase for components/classes) and be well-documented with JSDoc comments for functions, inline comments for complex logic, and README files for each module.

		2. System should use modular architecture with clear separation of concerns (MVC pattern: Models, Controllers, Routes separated), reusable components, and loosely coupled modules for easy maintenance and updates. 3. All API endpoints should be documented using Swagger/OpenAPI specification with endpoint descriptions, request/response examples, required parameters, authentication requirements, and error responses. 4. System logs should capture all critical events (user registrations, payments, errors), errors with full stack traces, security events (failed logins, suspicious activities), and admin actions (campaign approvals, user suspensions) for debugging and audit purposes. 5. Database schema should be version-controlled with migration scripts for schema changes, allowing rollback to previous versions if needed, and documentation of schema changes with timestamps and reasons. 6. Code should maintain minimum 70% test coverage with unit tests for business logic (models, utilities), integration tests for API endpoints, and end-to-end tests for critical user flows (authentication, payment). 7. Deployment should use CI/CD pipelines (GitHub Actions) for automated testing on commit, code quality checks (linting, formatting), and automated deployment to staging/production with rollback

		capability.

5.8 Non-Functional Requirement 8: Availability

Table 19 Non-Functional Requirement Availability

ID	Category	Description
NF 8	Availability	1. System should be available 24/7 with scheduled maintenance during low-traffic hours (2-4 AM NPT, typically Wednesdays or Sundays) to minimize user impact. 2. Scheduled maintenance should be announced minimum 48 hours in advance via: platform banner notification, email to all active users, social media posts, and status page updates, with maintenance duration estimate provided. 3. Critical bugs (security vulnerabilities, payment failures, data loss issues) should be fixed and deployed within 24 hours of discovery, with hotfix process bypassing normal release cycle. 4. Load balancing should distribute traffic across multiple server instances if using multiple servers, with health checks every 30 seconds, automatic removal of unhealthy instances, and traffic rerouting to healthy servers. 5. Database failover mechanisms should ensure service continuity during primary server failures using MongoDB replica sets (minimum 3 nodes: primary, secondary, arbiter) with automatic failover completed

		within 60 seconds.

5.9 Non-Functional Requirement 9: Compatibility

Table 20 Non-Functional Requirement Compatibility

ID	Category	Description
NF 9	Compatibility	1. Frontend should be compatible with browsers released within last 2 years: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+, with graceful degradation for older browsers showing upgrade notification. 2. Mobile platform support: Android 8.0+ (released 2017) supporting 95%+ Android users, iOS 12+ (released 2018) supporting 98%+ iOS users, with responsive web design working across all screen sizes. 3. API should follow RESTful standards with consistent endpoint naming (/api/resource/), HTTP status codes (200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, etc.), JSON request/response format, and standard authentication headers. 4. Payment gateways should support multiple banks and financial institutions in Nepal (commercial banks, development banks, microfinance institutions) through eSewa and Khalti integration, covering 90%+ of Nepali banking users. 5. System should support integration with social media platforms for campaign sharing: Facebook Open Graph protocol for rich previews, Twitter Card

		metadata, LinkedIn sharing, and WhatsApp sharing on mobile. 6. File uploads should support common formats: images (JPEG, PNG, GIF, WebP), videos (MP4, MOV), documents (PDF) with automatic format conversion where beneficial (e.g., PNG to WebP for smaller size). 7. Internationalization support for future expansion: text externalized in language files (not hardcoded), date/time formatting using locale-aware libraries (Intl API), currency formatting (currently NPR, extensible to other currencies), and RTL (right-to-left) text support for potential languages like Arabic/Urdu.

6. Other Further Requirements

- Legal and Regulatory Requirements: The platform must comply with Nepal's core IT and financial regulations (IT Act 2000, NRB rules) and adhere to global data protection standards (similar to GDPR). It must enforce age verification (18+), protect intellectual property, and maintain detailed 7-year audit trails for financial transactions.
- Database Requirements: The system will use MongoDB Atlas with high availability (3-node replica sets) and enforce data security (AES-256 encryption for sensitive fields). Automated daily backups with defined retention policies and regular performance optimization are mandatory.
- Environmental Requirements: The system is required to be highly optimized for minimal resource consumption (efficient code, Gzip compression) and must support flexible deployment using Docker containerization in cloud or on-premises environments.

- Business Requirements: The platform's goal is to democratize funding for Nepali creators by prioritizing integration with local payment gateways (eSewa, Khalti) and offering a competitive 5% success-based commission. The infrastructure must be scalable (10,000+ campaigns) and support diverse campaign types.
- Support & Documentation Requirements: Comprehensive, multi-channel support (Email, Chat, Phone) and detailed documentation (Creator's Handbook, Video Tutorials, Admin Manuals) must be provided, along with a dedicated team to guide first-time creators.
- Disaster Recovery Requirements: A documented and tested Disaster Recovery plan is required, ensuring fast restoration with an RTO of 4 hours and minimal data loss (RPO of 1 hour) via offsite, encrypted backups.
- Integration Requirements: Core integrations include local payment gateways (eSewa, Khalti), cloud media management (Cloudinary), analytics (Google Analytics), and monitoring tools (Sentry, UptimeRobot).