HACK-A-LEAGUE 4.0

Project Proposal & Technical Blueprint

FinSaathi

AI-Powered Personal Finance Assistant for Underserved Communities

"Your Money. Your Future. Simplified."

Theme: Ignite the Future | Verticals: AI/ML · E-commerce & FinTech · Cybersecurity (RBAC)

Global Academy of Technology, Bengaluru · February 21–22, 2026

Executive Summary

Over 500 million Indians are either unbanked or underbanked. First-time earners, gig workers, daily-wage laborers, and rural users lack access to intelligent, personalized financial guidance. Existing fintech apps target the already-financially-literate urban demographic and are inaccessible due to complexity, language barriers, and the assumption of prior financial knowledge.

FinSaathi bridges this gap with a multilingual, AI-first personal finance assistant that speaks your user's language, literally. It combines a fine-tuned open-source LLM, real-time spending analytics, behavioral nudges, micro-investment recommendations, and a secure role-based access control architecture to serve four distinct user personas: End Users, Financial Advisors, Partner Institutions, and Admins.

Theme Alignment: 'Ignite the Future' — FinSaathi ignites financial futures for millions left behind by conventional banking and fintech, using cutting-edge AI to democratize wealth-building knowledge.

PART 1 — UI/UX Design Philosophy

FinSaathi is designed with a single, non-negotiable visual principle: premium should not mean exclusive. The interface must feel aspirational to a first-time smartphone user while remaining deeply intuitive. The design language is Liquid Glass — a contemporary aesthetic that Apple popularized and that signifies sophistication, transparency, and fluidity.

1.1 The Liquid Glass Design Language

Liquid Glass is a UI philosophy built on the interplay of translucency, depth, soft reflections, and smooth motion. Every surface in FinSaathi behaves as if it is made of frosted, slightly curved glass — catching ambient light, blurring content beneath it, and responding to touch with subtle elastic animations. The result is a UI that feels alive and regal without being cluttered or aggressive. The Royal theme — deep burgundy, warm gold, and crimson red against rich parchment white — evokes the gravitas of established wealth while remaining accessible to first-time users.

Core Visual Principles

- Glassmorphism Cards — All dashboard cards, modals, and bottom sheets use backdrop-filter: blur(24px) with a background of rgba(253,246,227,0.12) on dark burgundy surfaces and rgba(253,246,227,0.80) on light parchment surfaces. Cards have a 1px border with rgba(212,175,55,0.4) to simulate gold-edged glass.
- Layered Depth — The UI operates on a Z-axis concept. Background layers hold the ambient royal gradient mesh, mid-layers hold glass cards with gold-tinted edges, and top layers hold actionable controls. This creates a sense of palatial depth.
- Fluid Micro-Animations — All transitions use spring physics (stiffness: 300, damping: 20). Spending charts animate in with a draw effect; notification badges pulse in gold; bottom navigation icons morph on selection using Lottie animations.
- Ambient Gradient Mesh Background — The app background is a slowly shifting gradient mesh in deep burgundy (#1C0A00), dark gold (#B8860B), and royal crimson (#C0392B) — the FinSaathi Royal palette. This gradient animates at 0.3° per second creating a living, breathing canvas reminiscent of royal brocade.
- Typography — Two-font system: Cormorant Garamond (elegant serif with royal character) for headings and display text; Inter for body text. Both are open-source via Google Fonts. Font sizes follow an 8pt modular scale (12 / 14 / 16 / 20 / 24 / 32 / 40 / 56).
- Haptic Feedback — Every critical interaction (transaction confirmed, budget limit reached, savings goal hit) triggers a distinct haptic pattern on mobile, reinforcing the tactile premium feel.

Color System

Primary Burgundy
#1C0A00
App backgrounds, dark surfaces, navbars
Dark Gold
#B8860B
CTAs, active states, gold borders, chart accents
Royal Crimson
#C0392B
Alerts, badges, secondary actions, error states
Bright Gold
#D4AF37
Card border glow, icon highlights, dividers
Parchment White
#FDF6E3
Card backgrounds (light mode), primary surface
Ivory White
#FFFDF7
Alternating table rows, input backgrounds
Surface Dark Glass
rgba(28,10,0,0.75)

Glass card background on dark surfaces

Surface Light Glass

rgba(253,246,227,0.80)

Glass card background on light surfaces

Text Primary

#2C0A0A

Primary text on light parchment surfaces

Text Secondary

#6B4F3A

Captions, metadata, placeholder text

Gold Border

#D4AF37

Card edges, table borders, input outlines

Warning

#C0392B

Budget alerts, overspend indicators

Success

#7D6008

Goal achieved, positive balance (warm gold-green)

1.2 Web Application (Responsive PWA)

The web app is built as a Progressive Web App (PWA) using React 18 + Vite, installable on Android and iOS without an App Store. It adapts fluidly from 320px mobile widths to 4K desktop displays using a 12-column CSS Grid system.

Dashboard Layout (Desktop)

- Left Sidebar (240px) — Deep burgundy (#1C0A00) glass-surface navigation panel with collapsible labels. Gold (#D4AF37) active indicator line on selected items. Contains avatar with gold ring, role badge in crimson, and primary nav items with Lottie icon animations.
- Main Content Area — Three-column bento grid at 1280px+ breakpoint. Cards are parchment-white glassmorphic surfaces with box-shadow: 0 8px 32px rgba(212,175,55,0.20) — a subtle gold ambient glow.
- Right Panel (320px) — AI Chat Assistant drawer with parchment (#FDF6E3) background and gold-bordered input. Slides in from the right over the main content with a warm-tinted glass blur overlay.
- Top Bar — Frosted parchment glass strip (blur: 20px, background: rgba(253,246,227,0.85)) with greeting in Cormorant Garamond, global search with gold focus ring, crimson notification bell badge, and dark/light mode toggle.

Dashboard Layout (Mobile)

- Bottom Navigation Bar — Deep burgundy glass bar with gold-highlighted active tab. Gold line indicator above active icon. Center FAB is a gold (#D4AF37) circle with crimson icon for quick expense entry.
- Swipeable Card Stack — Spending summary, savings goals, and AI insights presented as horizontally swipeable parchment-glass cards with gold-border edges at the top of the home screen.
- Pull-to-Refresh — Custom liquid gold ripple animation from pull point replaces the standard spinner.

- Bottom Sheet Modals — Parchment white bottom sheets with a gold drag handle pill and crimson close button. Spring animation on open/dismiss.

1.3 Mobile Application (React Native)

The mobile app shares the same design system and component library as the web app via a shared Storybook-documented component package. Platform-specific implementations:

- iOS — Uses SwiftUI BlurView bridged via React Native for native-level glass effects on parchment surfaces. Follows Human Interface Guidelines for gesture handling.
- Android — Uses RenderEffect API (Android 12+) for backdrop blur over the royal burgundy base. Falls back to a semi-transparent parchment overlay with subtle gold-noise texture for Android 11 and below.
- Shared — Reanimated 3 for all animations, React Navigation 6 with custom gold-accent glass tab bar, Skia canvas for chart rendering in gold/crimson palette.

1.4 Onboarding Experience

First impressions determine retention, especially with first-time users. FinSaathi's onboarding is designed as a conversational flow, not a form:

- Screen 1 — Animated splash: royal burgundy background with a slowly rotating gold gradient mesh. FinSaathi logo in Cormorant Garamond fades in with a golden shimmer effect.
- Screen 2 — Language selection presented as large, tactile gold-bordered glass chips on parchment white (12 Indian languages supported from day one).
- Screen 3 — WhatsApp or Mobile Number sign-in. Input field with gold focus ring and crimson CTA button. No email required, reducing friction for non-technical users.
- Screen 4 — AI-guided 3-question financial profiling: income range (gold gradient slider), primary goal (crimson-active chips), existing savings (toggle). No intimidating forms.
- Screen 5 — Personalized dashboard generated instantly. The user sees their AI advisor's first message in a gold-bordered chat bubble waiting for them.

PART 2 — Feature Architecture by User Role

FinSaathi operates on a four-tier Role-Based Access Control (RBAC) model. Each role has strictly scoped permissions managed by a middleware layer on the backend. The RBAC implementation uses the casbin open-source library with a policy.csv model stored in PostgreSQL.

RBAC Architecture: Subject (Who) → Resource (What) → Action (How). Policies are enforced at the API gateway level using casbin-node middleware. No role can escalate its own privileges. All permission changes are audit-logged.

END USER
Low-income individual, gig worker, first-time earner. Core beneficiary of the platform.
FINANCIAL ADVISOR

Verified human advisor assigned to a cohort of users. Read-only on user data unless user grants write consent.
PARTNER INSTITUTION
NGOs, microfinance institutions, banks onboarded as data/product partners.
ADMIN
Platform operator. Full system access with mandatory 2FA and audit trail.

2.1 End User — Feature Set

2.1.1 AI-Powered Chat Financial Advisor (Core Feature)

The heart of FinSaathi is a conversational AI built on a fine-tuned Mistral-7B-Instruct model, hosted locally or via Ollama on the backend server. The model is fine-tuned on Indian personal finance data to understand local context: SIPs, PPF, chit funds, MSME lending, UPI fraud patterns, and regional income categories.

- Model: Llama 8B (Apache 2.0 License) — fine-tuned using QLoRA on a custom dataset assembled from RBI guidelines, SEBI investor education material, and synthetic Q&A pairs generated using GPT-4o for bootstrap (then distilled).
- Fine-tuning Framework: Hugging Face PEFT + TRL (SFTTrainer). Training run on Google Colab A100 (free tier) or Kaggle GPU. Model weights saved to Hugging Face Hub (free).
- Serving: Ollama (open-source, MIT license) on the backend server. Supports streaming token output for real-time chat feel.
- MCP (Model Context Protocol): Implemented using Anthropic's open-source MCP specification. The LLM is given structured tool calls — get_user_balance, get_recent_transactions, get_investment_portfolio, get_market_data. The model decides which tools to call based on the user's query, retrieves live data, and synthesizes a grounded, factual response. This prevents hallucination of financial figures.
- RAG Pipeline: User's financial documents (salary slips, bank statements uploaded as PDF) are chunked, embedded using sentence-transformers/all-MiniLM-L6-v2 (free, HuggingFace), and stored in a Chroma vector database (Apache 2.0). The LLM retrieves relevant chunks before answering queries about the user's own financial situation.
- Multilingual Support: IndicTrans2 (open-source, AI4Bharat) translates user input to English before the LLM processes it, then translates the response back. Supports Hindi, Kannada, Tamil, Telugu, Bengali, Marathi, and 5 more.

2.1.2 Intelligent Expense Tracking

- SMS Parsing: An on-device NLP model (fine-tuned DistilBERT, 66MB) reads bank SMS notifications with user permission. Extracts merchant, amount, category, and timestamp. No raw SMS ever leaves the device.
- UPI Deep-Link Integration: Detects UPI payment completions from GPay, PhonePe, Paytm via Android intent listeners and iOS URL scheme callbacks. Auto-logs transactions.
- Manual Entry with AI Assist: User types or speaks a transaction ('spent 200 on chai and samosa at the dhaba'). A Named Entity Recognition model extracts: amount=200, category=Food, merchant=dhaba. One-tap confirmation.
- Receipt OCR: Camera capture of physical receipts. Parsed using Tesseract OCR (Apache 2.0) + a post-processing DistilBERT model for field extraction (total, merchant,

date, items).

- Auto-Categorization: XGBoost classifier trained on 50,000+ Indian transaction descriptions from open datasets. 23 expense categories including Autorickshaw, Festival, Pooja, and EMI — categories relevant to Indian users.

2.1.3 AI Budget Planner & Spending Insights

- 50/30/20 Adaptive Rule: The AI adjusts the classic budgeting rule based on the user's income tier. For users below ₹15,000/month, it applies a 60/20/20 rule (more for needs). Explained in plain language with local examples.
- Anomaly Detection: Isolation Forest model (scikit-learn) flags unusual spending — e.g., 'You spent 3x your usual food budget this week. Was there a special occasion?' This doubles as a soft fraud detection layer.
- Spending Forecast: Prophet (Meta, open-source) time-series model forecasts month-end spending based on first-week patterns. Displayed as a glass-morphic projection chart with confidence intervals.
- Category Drill-Down: Tap any spending category to see an itemized glass card list with merchant logos (fetched from Clearbit Logo API, free tier), amounts, and AI-generated observation (e.g., 'You visit this medical store every 2 weeks. Consider buying in bulk to save ₹180/month.').

2.1.4 Savings Goal Engine

- Goal Templates: Pre-built goal templates for Indian life events — Wedding, Home Down Payment, Child's Education, Emergency Fund, Festival Shopping. Each has a localized AI-generated savings plan.
- Micro-Savings Automation: Round-up feature: every UPI transaction is rounded up to the nearest ₹10, and the difference is swept to a savings pocket. User can set rule: 'Save ₹5 every time I spend on food delivery.'
- Gamification Layer: Savings streaks, milestone badges (Gold-foil glass badge cards animate in with a crimson confetti particle burst), and a community leaderboard (opt-in, anonymized). Behavioral economics nudges — loss aversion framing ('You're ₹500 away from losing your 30-day streak') proven to improve savings adherence.
- Goal Progress Visualization: Circular progress ring with liquid fill animation (Skia canvas). Color transitions from crimson red at 0–40% to bright gold at 100%, symbolizing the journey from struggle to royalty.

2.1.5 Micro-Investment Recommendations

- Risk Profiling: 6-question psychometric quiz (inspired by SEBI's investor suitability framework) generates a Conservative / Moderate / Aggressive profile stored in the user model.
- Recommendation Engine: Collaborative filtering model (Surprise library, open-source) trained on anonymized investment behavior data combined with content-based filtering on fund attributes. Recommends: Direct Mutual Funds (via MF API from MFAPI.in — free), Digital Gold (via MMTC-PAMP public API), NSC/PPF/Sukanya Samriddhi (Government schemes, static data).
- SIP Calculator: Visual SIP calculator with AI-generated plain-language explanation of compounding ('If you save ₹500/month for 10 years at 12%, you'll have ₹1,16,000. That's your child's first year of college fees.').
- Zero-Commission Direct Funds: The platform links directly to AMFI

(Association of Mutual Funds in India) registered fund houses via their public APIs. No commissions, no middlemen.

2.1.6 Financial Literacy Hub

- Bite-sized daily lessons (2–3 minutes) on financial topics: What is inflation? How does EMI work? What is CIBIL score? Content generated by the fine-tuned LLM, reviewed by advisors.
- Lessons adapt to the user's learning pace. If a user doesn't engage with a topic for 3 days, the AI sends a WhatsApp/SMS nudge (via Twilio free tier or MSG91).
- Quiz system with streak rewards. Completing 5 lessons unlocks a 'Financial Health Certificate' shareable on LinkedIn.
- Audio mode: All lessons available as text-to-speech using Coqui TTS (open-source, Mozilla Public License). Critical for low-literacy users.

2.1.7 Document Vault

- Encrypted storage for PAN card, Aadhaar (last 4 digits only, tokenized), salary slips, bank statements, insurance policies.
- Documents are encrypted client-side using AES-256-GCM before upload (Web Crypto API in browser, Expo SecureStore on mobile).
- AI auto-extracts key fields from uploaded documents and pre-fills forms (KYC, loan applications).

2.2 Financial Advisor — Feature Set

Financial Advisors are verified professionals (NISM-certified or NGO-affiliated counselors) assigned to cohorts of 50–200 users. Their RBAC permissions are read-heavy; all write actions require explicit user consent stored as a time-limited consent token.

2.2.1 Advisor Dashboard

- Cohort Overview: Glass-card grid showing all assigned users with a color-coded financial health score (0–100, computed by the ML health scoring model described below). Red = critical intervention needed, Yellow = at-risk, Green = on-track.
- Financial Health Score Model: Multi-feature regression model (XGBoost) trained on features: savings rate, debt-to-income ratio, emergency fund coverage, investment diversification, spending volatility, and goal adherence. Output is a 0–100 score with plain-language explanation.
- Alert Queue: AI surfaces users who need immediate attention — e.g., 'User X has not made any transactions in 15 days (possible income loss)' or 'User Y has exceeded their food budget for 3 consecutive months.'
- Bulk Communication: Send personalized financial tips to the entire cohort. The AI personalizes the template for each user based on their profile (income, goals, risk level) before sending via WhatsApp/SMS.

2.2.2 User Detail View (With Consent)

- Full spending breakdown, savings trajectory, and investment portfolio — all in the same glass-morphic UI as the user's view, but with an additional advisor annotation layer.
- Advisor can leave time-stamped notes on a user's profile (not visible to the user unless shared).

- Suggested Action Panel: AI generates 3 ranked intervention suggestions for each user the advisor is viewing ('Suggest an emergency fund goal', 'Recommend SIP amount reduction given recent income dip', 'Flag potential EMI overload').

2.2.3 AI Co-Pilot for Advisors

- The same fine-tuned Mistral LLM serves the advisor but with a different system prompt granting access to aggregated anonymized cohort insights.
- Advisor can ask: 'What is the most common financial mistake in my cohort this month?' or 'Which users are closest to their savings goals and could benefit from a motivational nudge?'
- AI drafts personalized action plans for individual users that the advisor reviews and approves before delivery.

2.2.4 Educational Content Curation

- Advisors can create custom lesson modules for their cohort using a block-based editor (similar to Notion).
- AI suggests relevant content blocks based on the cohort's common financial pain points detected in the past 30 days.
- Content goes through an admin review queue before publishing.

2.3 Partner Institution — Feature Set

Partner Institutions (banks, MFIs, NGOs, government schemes) can integrate with FinSaathi via a secure API to offer their products/services to matched users. Their RBAC access is limited to anonymized aggregate data and their own API endpoints. They cannot access individual user PII.

2.3.1 Product Listing & Matching

- Partners list financial products (microloans, insurance, savings accounts, government schemes) via a structured form in the Partner Portal.
- FinSaathi's recommendation engine matches products to users based on eligibility criteria (income, location, credit score tier, employment type) without sharing user PII with the partner.
- Matching uses a privacy-preserving federated scoring approach: the eligibility check runs on FinSaathi's servers and returns only a binary match/no-match signal to the partner.

2.3.2 Aggregate Analytics Dashboard

- Partners see aggregated, k-anonymized (k≥50) analytics about their matched user cohort: income distribution histogram, geographic heat map (district level, not village level), product uptake funnel.
- All analytics are computed using differential privacy (Google's DP library, open-source) before serving to partners. No query can reconstruct an individual.
- Partners can request A/B tests for their product messaging, run by FinSaathi's admin team.

2.3.3 API Integration Suite

- REST API with OAuth 2.0 (PKCE flow) for partner authentication.
- Webhooks for real-time events: user_matched, user_applied, user_onboarded.
- Sandbox environment with synthetic data for integration testing.
- API rate limiting per partner tier enforced via Redis token bucket.

2.4 Admin — Feature Set

Admins are platform operators. Admin access requires mandatory hardware TOTP 2FA (FIDO2/WebAuthn via open-source SimpleWebAuthn library). All admin actions are logged to an immutable audit trail stored in an append-only PostgreSQL table with row-level checksums.

2.4.1 User & Role Management

- Full CRUD on user accounts, advisor assignments, and partner onboarding.
- Role assignment uses casbin policy management UI — admins select subject, resource, and action from dropdowns; raw policy strings are never manually edited.
- Bulk user import via CSV for partner-onboarded cohorts (e.g., an NGO uploads a list of 500 beneficiaries).

2.4.2 Model Monitoring & Retraining

- MLflow (open-source) dashboard embedded in the admin panel. Shows model performance metrics: recommendation click-through rate, chat satisfaction score (thumbs up/down), categorization accuracy, savings goal completion rate.
- Data drift detection using Evidently AI (open-source). Alerts admin when the distribution of incoming transaction data deviates significantly from training data, triggering a retraining pipeline.
- One-click retraining: Admin triggers a Celery background job that pulls fresh data, fine-tunes the model using PEFT/QLoRA, evaluates on a holdout set, and if performance improves, deploys the new weights to Ollama.

2.4.3 Content Moderation & Quality Control

- All AI-generated financial advice is logged. A secondary classifier (fine-tuned on Indian financial regulation compliance rules) flags potentially non-compliant advice for human review.
- Advisor-created content sits in a review queue before publishing. Admin can approve, edit, or reject.
- Abuse detection: Unusual API call patterns from partner accounts trigger automatic rate-limit escalation and alert to admin.

2.4.4 Platform Analytics

- Real-time dashboard built with Apache Superset (open-source) showing: DAU/MAU, feature adoption heatmap, average financial health score over time, savings mobilized (aggregate ₹), investment recommendations accepted.
- Cohort analysis to measure long-term impact: Do users who engage with the Financial Literacy Hub improve their savings rate within 90 days?
- Exportable reports for impact reporting to investors, government bodies, and grant applications.

Data Sources — Free & Open

MFAPI.in (mfapi.in)
All Indian mutual fund NAVs — Free, no auth required
RBI DBIE (data.rbi.org.in)
Interest rates, inflation, banking statistics — Government Open Data
NSE India Data API (nseindia.com)
Stock market data for investment context — Free (scraping permitted)
PMJDY Dashboard (pmjdy.gov.in)
Financial inclusion metrics for model validation — Government Open Data
Hugging Face Datasets: financial_phrasebank
Sentiment classification training data — CC BY-SA 3.0
Hugging Face: ai4bharat/IndicTrans2
Translation model for 22 Indian languages — MIT License
Open Government Data (data.gov.in)
Income, employment, and demographic data by district — NGOAL
SEBI Investor Education PDFs
Fine-tuning corpus for financial Q&A — Public domain
World Bank Financial Inclusion Dataset (FINDEX)
Benchmark data for underserved community modeling — CC BY 4.0
Synthetic Data Generator: SDV (open-source)
Generate privacy-safe training transactions — MIT License

Full Technology Stack — 100% Free & Open Source

5.1 Frontend

Framework
React 18 + Vite
MIT
Mobile
React Native (Expo)
MIT
Animations
Framer Motion + Reanimated 3
MIT
Charts
Recharts + React Native Skia
MIT
State Management
Zustand
MIT
API Client

TanStack Query (React Query)
MIT
UI Components
Radix UI (headless) + custom glass CSS
MIT
Fonts
Satoshi + Inter (Google Fonts)
OFL
Lottie Animations
lottie-react / lottie-react-native
Apache 2.0
PWA
Vite PWA Plugin + Workbox
MIT
i18n
i18next + react-i18next
MIT

5.2 Backend

Runtime
Node.js 20 LTS + Fastify
MIT
Language
TypeScript
Apache 2.0
ORM
Prisma
Apache 2.0
Primary Database
PostgreSQL 16
PostgreSQL License
Cache / Queue
Redis 7 + BullMQ
BSD / MIT
RBAC
Casbin (node-casbin)
Apache 2.0
Auth
Passport.js + SimpleWebAuthn (FIDO2)
MIT
File Storage
MinIO (S3-compatible)
AGPL-3.0
API Gateway
Kong OSS
Apache 2.0
Search
MeiliSearch
MIT
Background Jobs
Celery (Python) for ML pipelines
BSD

5.3 AI / ML Layer

Base LLM
Llama 8B
Apache 2.0
LLM Serving
Ollama
MIT
Fine-tuning
Hugging Face PEFT + TRL (QLoRA)
Apache 2.0
MCP Implementation
Anthropic MCP SDK (open-source)
MIT
Embeddings
sentence-transformers/all-MiniLM-L6-v2
Apache 2.0
Vector DB
ChromaDB
Apache 2.0
Classical ML
scikit-learn + XGBoost
BSD / Apache 2.0
Time Series
Prophet (Meta)
MIT
NLP / NER
spaCy + fine-tuned DistilBERT
MIT / Apache 2.0
OCR
Tesseract OCR
Apache 2.0
Translation
IndicTrans2 (AI4Bharat)
MIT
TTS
Coqui TTS
MPL-2.0
Model Tracking
MLflow
Apache 2.0
Data Validation
Evidently AI
Apache 2.0
Privacy
Google's DP Library
Apache 2.0
Recommendation
Surprise library
BSD

5.4 DevOps & Infrastructure

Containerization
Docker + Docker Compose
Apache 2.0
Orchestration
Kubernetes (k3s for local)
Apache 2.0
CI/CD
GitHub Actions
MIT
Monitoring
Grafana + Prometheus
AGPL / Apache 2.0
Log Management
OpenTelemetry + Loki
Apache 2.0
Analytics
Apache Superset
Apache 2.0
Secrets
HashiCorp Vault
BSL (free for self-hosted)

MCP Architecture — Deep Dive

Model Context Protocol (MCP) is used as the bridge between the fine-tuned Mistral LLM and FinSaathi's live data layer. Rather than stuffing the entire user profile into a single prompt (expensive and imprecise), the LLM uses structured tool calls to retrieve exactly the data it needs to answer a query.

MCP Tool Registry

get_user_profile
Returns income tier, risk profile, goals list, language preference
get_recent_transactions
Returns last N transactions with category, amount, merchant, date
get_budget_status
Returns current month spend vs budget per category
get_savings_goals
Returns all active goals with progress percentage and projected completion
get_investment_portfolio
Returns holdings: SIP details, gold grams, scheme NAV, total value
get_market_data
Returns current Sensex, Nifty, gold price, FD rates from NSE/MFAPI
get_financial_health_score
Returns the ML-computed health score with feature attribution
search_schemes

Searches government scheme database by eligibility criteria

get_rag_context

Retrieves relevant chunks from user's uploaded documents (ChromaDB)

Example interaction flow: User asks 'Should I increase my SIP amount?' → LLM calls get_budget_status, get_savings_goals, get_investment_portfolio, get_market_data → synthesizes response: 'Based on your current budget, you have ₹1,200 unspent each month. Your retirement goal needs ₹800 more per month to be on track. Current market conditions (Nifty at 52-week high) suggest a ₹500–800 SIP increase is reasonable. Want me to show you which fund to add it to?'

24-Hour Hackathon Execution Plan

Hour 0–2

Environment setup, repo init, DB schema finalized, Ollama + Mistral running locally

Hour 2–5

RBAC middleware live, Auth (OTP login) working, Core DB models seeded

Hour 5–9

End User dashboard UI (glassmorphic), Expense tracking (manual + SMS parse), Budget charts

Hour 9–13

AI Chat (MCP + Ollama) functional, RAG pipeline with ChromaDB working

Hour 13–17

Savings Goals engine, Micro-investment recommendation engine, Multilingual (Hindi + 1 regional)

Hour 17–20

Advisor dashboard, Financial Health Score model, Partner portal skeleton

Hour 20–22

Admin panel, MLflow integration, Polish animations, Mobile responsiveness

Hour 22–24

Bug fixes, demo script preparation, presentation deck, GitHub README + demo video

Alignment with Judging Criteria

Clarity of Concept

Single sentence pitch: AI finance assistant for India's 500M underserved. Every feature ties back to this.

Originality & Novelty

MCP + fine-tuned local LLM + multilingual RAG for personal finance is not done in any existing Indian fintech app.

Real-World Value

Directly addresses financial exclusion, a documented crisis. Zero-commission investment access creates measurable user benefit.

Technical Excellence

QLoRA fine-tuning, MCP tool orchestration, federated privacy, RBAC via casbin, differential privacy — strong technical depth.

Scalability
Stateless Fastify backend, Redis caching, Ollama horizontally scalable, PWA removes App Store dependency.
Innovation
Combining MCP + local LLM + Indian language translation + behavioral nudges in a unified platform is genuinely novel.
Presentation
Liquid Glass UI ensures the demo looks unlike any other hackathon project — memorable and premium.
Team Collaboration
Clear module ownership: UI Lead, ML Lead, Backend Lead, Data/Pitch Lead.
Project Completion
24-hour plan prioritizes a working core (Chat + Expense + Budget) with progressively added features.

FinSaathi — Ignite the Future

Global Academy of Technology · Hack-A-League 4.0 · February 21–22, 2026
hackaleague@gmail.com · Built with 100% Open-Source Tools