Co-creation HUB

Annex A: Terms of Reference

About the GATEWAY Program

Gig work provides opportunities for young people, particularly young women, to access income-earning opportunities and alleviate poverty with flexible and adaptable career opportunities. The GATEWAY Program supports youth in developing digital skills and accessing digital creative gig work across multiple sectors. To ensure the success of gig workers in Nigeria, the Program will provide training, mentorship, access to work tools, gig placement assistance, and access to enhanced financial services.

Location: The Program will be deployed across ten (10) Nigerian states/locations: Abuja (FCT), Enugu, Delta, Edo, Kaduna, Kano, Lagos, Ogun, Oyo, and Rivers. The selected locations are areas with highly marginalised population groups, including internally displaced people's camps, and have robust internet connectivity.

The Program aims to accelerate and advance the skill set and gig work opportunities for youth, creating sustainable, dignified, and fulfilling jobs and improved livelihoods for young people in Nigeria.

Program Participants

The Program shall target two types of financially disadvantaged young women/youth: (i) Unskilled Group – The "unskilled" group of Program Participants shall consist of at least 100,000 financially disadvantaged young women (age 18 to 35) who:

(A) have no specific digital creative skills but possess fundamental literacy, numeracy, and general digital capabilities, such as using internet search engines, sending and receiving emails, utilizing word processing software such as Microsoft Word, and managing files and folders on a computer;
(B) have at least secondary education as a minimum;
(C) are unemployed or engaged in low-skill, low-wage jobs and are seeking more stable and lucrative employment opportunities; and
(D) in the aggregate, consist of 100% young women, 5% persons living with disabilities, and 1% displaced youth.

(ii) Skilled Group – The "skilled" group of Program Participants shall consist of at least 400.000 financially disadvantaged young women/young individuals (age 18 to 35) who:

(A) have acquired relevant digital and creative skills through post-secondary education or vocational training, possibly in fields related to digital media, graphic design, or IT;
(B) are underemployed, freelancing without consistent work, or unaware of how to access gig work opportunities effectively;
(C) may or may not hold formal degrees; and
(D) in the aggregate, consist of 70% young women, 5% persons living with disabilities, and 1% displaced youth.

Each group of Program Participants shall include a subgroup of individuals who are unaware of available opportunities for income through gig work, especially in marginalized areas.

Overview of the GATEWAY Program Portal

To support the GATEWAY Nigeria program, this digital platform should provide end-to-end functionality covering the

1. pre-training phase (mass applications and onboarding),
2. training phase (learning, delivery, and engagement), and
3. post-training phase (gig placement and ongoing support).

The features below are organized by phase and include specific, actionable recommendations.

Open-source or cost-effective solutions are highlighted, and a **modular design** with third-party integrations is emphasized for scalability and extensibility.

The GATEWAY Nigeria digital platform should be feature-rich yet cost-conscious, covering the full lifecycle of participants from outreach to employment.

By utilizing a modular design with open-source components, the platform can provide application handling, massive-scale learning, vibrant community engagement, and seamless post-training support, including gig connections and financial linkages.

Each feature from gamified learning content to integrated micro-loan applications is designed to maximize the program's impact, empowering 400,000 youth with skills and connecting 340,000 of them to dignified gig work.

Pre-Training Phase: Application Management and Onboarding

Application Management Portal (1.5M Applicants)

- High-Volume Online Applications:

- A web-based application form capable of handling up to 1.5 million applicants.
- It should enable applicants to create accounts, save their progress, and submit the required information and documents.
- Validation gender, state of residence, and age, and disability, through submission of government-issued ID
- The form must be mobile-friendly and optimized for low-bandwidth use (given that many Nigerian youths access the internet via mobile).
- Automatically redirected to assessment

- Initial Skills Assessment: Online assessment tools to sort participants into the *unskilled vs. skilled* training tracks.
  - Diagnostic quiz or practical test to evaluate literacy / numeracy / basic digital skills, digital literacy and prior knowledge of digital creative skills (relating to specific tracks).
  - The platform should support timed quizzes, auto-grading of objective questions, and recording of scores.
  - This track sorting process ensures those with higher starting skills enter the skilled pathway while others receive foundational training.
  - All selections should be automatically done by the portal. No manual selection Based on the results, the system assigns the participant to the appropriate learning path.
    - [100% of women aged 18-35, who are unskilled, get automatically enrolled until we hit 100k completion
    - 100% of women aged 18 - 35 who are 'skilled' are automatically enrolled until we hit 280k completion
    - 100% of men aged 18-35 who are skilled are automatically enrolled until we hit 120k completion
    - 100% men and women aged 18 - 35 from each of the 10 states who meet other criteria, automatically enrolled until we hit state target for completion
    - 100% of PLWD or displaced should be enrolled, with no cutoff for completion
  - All who do not meet the selection criteria should be automatically disqualified and get auto-response

- Placement into Learning Pathway:
  - After assessment, the platform enrolls each participant in the correct curriculum track
    - Unskilled – Digital Marketing, Graphics Design, Video Production & Editing, and UIUX.
    - Skilled – General Gig Work Skills Training, Financial and Business Management Training,

- This should be automated to handle thousands of participants efficiently.

Participants receive a notification about their track and next steps (e.g., when courses start or if any bridging content is recommended before formal training).

- Applicant Database & Tracking:
  - A robust back-end to store applicant data securely and perform searches/filters.
  - Administrators should be able to sort and filter applications by specific criteria (such as location, demographics, education, etc.) to support quota targets (e.g., 80% women, 5% PLWD, as per program goals).
  - The system should include basic analytics (e.g., application count, dropout rates during form completion) and export capabilities for offline analysis and review.

- Automated Notifications:
  - Email and SMS notification tools to confirm application submission and communicate outcomes (putting people on unskilled vs skilled track, acceptance, waitlist, or rejection).
  - Given the scale, the platform should support bulk messaging and templated emails to applicants at various stages of the process.

- Admin Review & Selection Tools:
  - An administrator dashboard for managing the selection process.
  - Features should include workflow tools for application review (with multiple reviewers), batch approval or rejection, and the ability to flag candidates for follow-up.
  - Integration with simple analytics or AI could help identify top candidates or detect duplicate entries, but the priority is on simple, transparent filtering rules for fairness.

- Scalability & Cost:
  - To handle peak traffic (e.g., when applications open or close), consider a cloud-scalable architecture (load balancers, auto-scaling instances).
  - Using open-source web frameworks and databases (with optimization and indexing for large data) avoids licensing costs. For instance, a free applicant tracking system or a custom module built with open-source tools can manage the intake without per-user fees.
  - Static content (instructions, FAQs) can be served via CDN to reduce load on the main system.

Participant Onboarding & Assessment

- Account Creation and Profile Setup:
    - Accepted applicants seamlessly transition to participant accounts on the platform.
    - Their data from the application should populate a user profile.
    - They should set up a profile with personal details, a profile photo, and baseline information, such as skill interests – this profile will later serve as part of their learner/gig profile.

- Orientation Module:
    - An online orientation course for new participants.
    - This provides a welcome message, program overview, code of conduct, and a "how to use the platform" tutorial.
    - It can be a brief multimedia module that introduces key features of the platform and outlines program expectations.

- Documentation and Agreements:
    - As part of onboarding, the platform can collect any additional documents or agreements needed (e.g. consent forms, program contracts).
    - A simple e-signature feature or checkbox acknowledgments for terms and conditions should be included, so that formalities are completed digitally before training begins.

Training Phase: Learning Management System and Engagement

Learning Management System (LMS) Core Features

- Open-Source LMS Platform:
    - Deploy a proven open-source LMS to deliver content to ~400,000 learners. For example, Moodle or Open edX are ideal – Moodle is one of the most widely used open-source LMS (highly customizable), and Open edX (from MIT/Harvard) is designed for large-scale learning.
    - Using an open-source LMS avoids licensing fees while benefiting from a global community and plugin ecosystem. These platforms support multi-tenant or cohort-based setups, which can help segment users by batch or track.

- Course Content Delivery:
    - The LMS should support a rich variety of content: video lessons, slide presentations, readings, and interactive modules.
    - It must handle thousands of concurrent video streams (using a scalable content delivery network or integration with platforms like YouTube for cost-efficiency).
    - Interactive content standards like SCORM or xAPI should be supported so that external modules or content libraries can be integrated.
    - Course pages should be lightweight for low bandwidth, with options to download materials (PDFs, etc.) for offline use.

- Assessments and Quizzes:
    - Built-in assessment tools to conduct quizzes, assignments, and exams within courses.
    - This includes auto-graded quizzes (multiple-choice, drag-and-drop, etc.) for knowledge checks as well as assignment submission workflows for practical tasks (with grading rubrics if mentors/facilitators will review).
    - The platform should randomize question banks to prevent cheating at scale and allow time-limit settings.
    - Certificates or badges are awarded upon successful completion of key modules or at graduation from the program.

- Progress Tracking & Analytics:
    - Each learner should have a dashboard showing their course progress (% completion, scores, upcoming deadlines).
    - The LMS tracks detailed learning data – which lessons viewed, quiz scores, etc. – and provides this to both learners and administrators.
    - Mentors or instructors need access to reports to identify struggling learners (e.g. not logged in recently or failing quizzes) so they can intervene.
    - Program managers should be able to pull aggregate reports (completion rates, average scores, engagement metrics) to evaluate training effectiveness.
    - These analytics will feed into the program's M&E (monitoring & evaluation) – for example, tracking how many complete the unskilled vs. skilled pathways, in line with the 85% completion targets.

- Scalability & Performance:
    - The LMS must be highly scalable and tuned for performance.
    - This may involve deploying it in a distributed manner (multiple application servers, separate database servers, caching layers, etc.). Open edX, for instance, is built to be horizontally scalable and cost-effective for large online programs.

- We recommend using cloud infrastructure (AWS, Azure, or local cloud providers) to dynamically scale resources based on load. Caching and use of a CDN for static content (videos, images) will help maintain fast load times even as user count grows.
- Regular load testing should be conducted to ensure the system can handle peak usage, such as 100,000+ concurrent learners during a live session or deadline.

Learner Engagement & Community Tools

- Discussion Forums and Q&A:
  - A discussion forum integrated into the LMS where learners can ask questions, discuss course topics, and help each other.
  - Forums should be organized by course, by cohort, or by topic. This encourages peer learning and builds a sense of community despite the scale.
  - An open-source forum platform like *Discourse* could be integrated (for richer community features) or the built-in forum tool of the LMS can be used.
  - Moderation tools should be in place – e.g. the ability for staff or trained community moderators to remove inappropriate posts and ensure a safe, inclusive environment (important given a large youth cohort including women and vulnerable groups).

- Live Webinar/Virtual Classroom Integration:
  - For real-time engagement, the platform should integrate a webinar tool. Open-source options like *BigBlueButton* (which integrates with Moodle) allow live video classes, screen sharing, and breakout rooms. This can be used for weekly live coaching sessions or guest speaker events.
  - The system should support scheduling these live sessions and notifying learners.
  - Session recordings should be made available for those who cannot attend live (ensuring inclusivity across varied schedules).

- Messaging and Collaboration:
  - Enable direct messaging or group chats for learners and facilitators. This could be an in-app chat feature or integration with external messaging platforms (e.g. Mattermost or Rocket.Chat as open-source Slack alternatives).
  - Group collaboration spaces should be provided for project-based activities – for example, if learners are assigned team projects, they might get a private group workspace to chat and share files. These features foster teamwork and keep learners engaged with one another.

- Gamification:
    - Incorporate gamification elements to motivate learners.
    - The platform can award points or badges for completing modules, participating in forums, or helping others (e.g. a “Peer Helper” badge for answering questions).
    - Leaderboards should be used at a cohort level to spark friendly competition (while being careful to keep it healthy and not demotivating slower learners).

- Announcements & Notifications:
    - A centralized announcement system to post news, updates, or motivational messages.
    - Learners should see announcements on their dashboard and also get email/SMS alerts for important updates (e.g. new course available, maintenance downtime, or program milestone achievements). This keeps everyone informed and fosters a sense of progress and community (celebrating successes like “50,000 learners have completed Module 1!”).

Mentorship Program Support

- Peer Mentorship Matching & AI-Powered Mentorship Support:

The platform should include a mentorship module that enables **peer-to-peer mentorship** by matching participants with fellow learners based on shared interests, skill tracks, goals, or location. Participants will create profiles highlighting their backgrounds, skills, and learning objectives. Matching can be performed either manually by program staff or automatically through an algorithm that analyzes profile data. Participants can be paired in small peer mentorship groups or in one-on-one peer mentorship relationships, allowing for scalable, community-driven support across the 400,000 participants.

In addition to peer mentorship, the platform must feature an **AI-powered mentorship chatbot** that delivers personalized mentorship and advisory support. The chatbot should:

- Offer tailored recommendations, career advice, and learning pathways based on participant profiles, learning progress, and expressed goals.
- Provide instant responses to frequently asked questions about training programs, gig work opportunities, and career development.

- Guide participants through structured conversations on key personal and professional development topics.
- Suggest relevant peer mentors, resources, and platform activities based on user needs.

- Mentorship Interaction Tools:

Participants will have access to integrated communication tools designed to support mentorship interactions. These tools should include:

- Direct messaging and group chat functionalities for peer mentorship groups.
- A scheduling feature with calendar integration, allowing participants to organize virtual meetings or collaborative sessions.
- Dedicated spaces for peer mentors and mentees to collaborate, share updates, and exchange ideas.

Participants will also be encouraged to log notes, reflections, or session summaries after each interaction (including AI chatbot interactions), supporting personal accountability and continuous learning.

- Goal Setting and Tracking:

The platform should offer tools for participants to collaboratively set and track developmental goals. Examples include goals like “Complete XYZ project by next month” or “Improve interview skills.”
Participants can use a simple goal-tracking tool or checklist to monitor progress, with options to update goal statuses, add comments, and celebrate milestones. This encourages accountability and keeps mentorship engagements focused and purposeful.

- Resource Sharing:

Participants and peer mentors should have access to resource-sharing tools within the platform. These may include:

- Shared folders for documents, articles, and tools.
- Private discussion threads or message boards where participants can post tips, resources, and useful links relevant to their mentorship discussions.

- Monitoring and Feedback Mechanisms:

To ensure quality mentorship experiences, program administrators need visibility into mentorship activities. The platform should:

- Log all mentorship interactions (peer and AI chatbot sessions).
- Allow participants to provide feedback or rate their peer mentorship experiences.
- Capture data on mentorship engagement levels and satisfaction to help program staff identify active contributors and those needing additional support.
- Support recognition and reward mechanisms for highly engaged peer mentors.

- Integration with LMS & Analytics:

The mentorship module must integrate seamlessly with the Learning Management System (LMS) and other learning tools. Key features should include:

- Access for peer mentors to view their mentees' learning progress (e.g., course completions, quiz scores) to offer better guidance.
- Notifications for mentors and participants about relevant learning milestones, such as inactivity or performance drops.
- Analytics dashboards for program administrators to track mentorship outcomes, learning engagement, and participant progression across both peer mentorship and AI-powered interactions.

Community and Networking

- Peer Community Forums:
  - Beyond course-specific forums, an open community forum for all participants and alumni will facilitate broader networking opportunities. Here, members can discuss general topics: gig economy tips, success stories, challenges, regional meetups, etc.
  - Sub-forums should be created for special interest groups (e.g. women in tech, creatives, freelancers in a specific region, PLWD support). This persistent community space will remain valuable during and after training (forming the alumni network).

- Social Media Integration:
  - While the platform itself should host the core community features, integrating popular social channels will help with engagement.
  - Essential announcements or content could be cross-posted to a private Facebook group or Telegram channel, as many Nigerian youth use these.
  - It should have a feature for moderation and privacy
  - At minimum, provide easy share options so participants can celebrate their new certificates or project completions on LinkedIn, Twitter, etc., which also boosts

program visibility.

- Expert Webinars:
    - We will host periodic live webinars or AMA (Ask Me Anything) sessions with industry experts, successful gig workers, or program alumni. These live events (using the integrated webinar tool or even streaming via YouTube Live) create excitement and give learners exposure to real-world insights.
    - The platform should facilitate registration for these events and collect questions in advance. Recordings are then stored in a resource library for those who missed it.

- Content Library & Continuous Learning:
    - Provide a repository of additional learning materials and links beyond the formal courses. This library will include articles on gig economy best practices, advanced tutorials, templates (e.g., how to create a freelancer profile, how to write a proposal, etc.), and links to partner resources.
    - We will curate this library with open educational resources to keep costs down. It remains a reference for participants even after they finish the formal training.

- Community Moderation & Safeguarding:
    - Given the large user base, community management features are crucial.
    - Implement user reporting functions (so participants can flag inappropriate behavior/posts), and have moderation workflows for staff.
    - Clearly post community guidelines or a code of conduct. This is particularly important for inclusivity – the community should be safe for young women and vulnerable groups.
    - Admin tools to remove or ban abusive users, and possibly AI-assisted content moderation for flagging harassment keywords, will help maintain a positive environment.

Post-Training Phase: Gig Placement and Financial Enablement

Gig Placement Support

- Gig Opportunities Portal:
    - Integrate a gig platforms within the platform where partners and local companies can post gig opportunities targeted at program graduates. This acts as a curated marketplace for participants.
    - Rather than duplicating their listings, the portal can aggregate or showcase opportunities from major gig platforms (via RSS feeds or APIs if available) that are

relevant to the skills taught.

- Integration with Global Gig Platforms:
    - The platform should guide and streamline the process of getting participants onto major gig economy platforms (like Upwork, Fiverr, Freelancer, etc.).
    - This should include deep links or embedded widgets to help create profiles on those external platforms.
    - If possible, secure API integrations – for example, using any available partner APIs to pre-fill profile data or display a feed of recommended gigs.
    - *Global gig platform partnerships are a program goal*, such as collaborating with platforms to enable local payment options for Nigerians.
    - The system can facilitate this by directing users to those platform features and providing support (e.g. tutorials on how to link a local bank or fintech wallet to Upwork).
    - While full technical integration might be limited by the platforms themselves, even a single sign-on or data exchange (like verifying training certificates to display on a gig profile) could add value.

- Profile Building and Portfolio:
    - Within the platform, allow graduates to build a gig portfolio showcasing their skills and accomplishments.
    - This feature lets users assemble evidence of their work (course certificates, project work, badges earned, etc.) into a profile that they can make public or export.
    - After completing training, a participant's profile should show a summary of skills learned and a link to download their certificate.
    - They will be encouraged to use this information on external gig platforms. The platform should offer a standardized CV or LinkedIn profile export tool – integrating with LinkedIn's API to auto-populate a user's LinkedIn profile with their new skills/certifications (making it easier for them to be noticed by employers).

- Career Guidance and Job Search Skills:
    - Provide resources or mini-courses on how to succeed in the gig market. This includes modules on creating strong proposals, communication with clients, time management for freelancers, etc. These will be part of the post-training curriculum.
    - Additionally, mentorship will continue here: mentors or career coaches could review participants' profiles or proposals and give feedback.
- Tracking Employment Outcomes:

- Include a feature for graduates to report when they secure a gig or job, and log basic details (platform, job type, income range).
- This can be as simple as a form or survey that periodically asks alumni about their employment status. Tracking this within the platform helps measure the program's success (aiming for 340,000 youth securing gig work within 6 months of completion).
- Where possible, integrate with gig platform data – for example, if a partner gig platform is willing to share referral data or earnings of program graduates (via an API or data sharing agreement), the system could automatically update a user's status.
- However, privacy and data agreements need to be managed carefully. At minimum, self-reported outcomes and perhaps verification by mentors can be used to gather placement data.

Financial Services Integration

- Micro-Loans for Work Tools:
  - Many participants will need equipment (laptops, internet devices) to start gig work. The platform should facilitate access to micro-loans provided by partner credit providers.
  - Concretely, after a participant completes all training modules and meets the eligibility criteria (as defined with the microloan provider), the system can flag them as "Eligible for Work-Tool Microloan".
  - Features could include: a notification or dashboard banner for eligible users, an in-platform form to apply for the loan, and document upload for KYC (ID, proof of residence, etc.).
  - The platform can then securely transmit the application to the microloan provider's system (via an API or even automated email if API is not available).
  - Status updates (approved, funds disbursed) can be fed back into the user's dashboard. By integrating this process, the platform reduces friction for participants – e.g. automatically confirming through data that the user has completed training (a requirement for the loan) so they don't have to prove it separately.

- Insurance & Pension Products:
  - We will work with partner financial service providers (FSPs) to offer micro-insurance, health insurance, and portable pension plans tailored to gig workers.
  - The platform should host a "Financial Services" section where users can see all available partner offers (loans, insurance, savings plans) and click through to enroll.

    - The platform should introduce these options during or after training. For example, upon program completion, present users with information about an optional gig worker insurance plan (covering illness or equipment loss) and a micro-pension scheme where they can contribute voluntarily.
    - Features to support this will be:
        - informational pages and calculators (to illustrate benefits),
        - links or embedded sign-up forms from the FSPs, and possibly an API integration that lets users sign up using their platform profile data to pre-fill forms.
        - The integration might simply redirect to the provider's app/website for the final enrollment, but the key is to centralize access and awareness on the training platform. This aligns with program outputs to connect graduates to enhanced financial services like insurance and pensions.

- Financial Literacy & Tools:
    - Related to the LMS content on this, simple tools like a budgeting spreadsheet template or income tracker will be provided.
    - If technically feasible, integrate a third-party budgeting app or API that helps users track their gig earnings and set aside savings for pension or loan repayments. These tools drive home the value of financial planning alongside the services offered.

- Payment Integration:
    - For any stipends or rewards given during training an integration with a mobile money or payment gateway (Flutterwave, Paystack, etc.) could allow disbursement directly through the platform.

Alumni Network & Continuing Engagement

- Alumni Community Access:
    - After training, participants retain access to the platform community. The same discussion forums and groups transition to serve as alumni networks.
    - An Alumni tag or section should be created to differentiate those who have completed training.
    - The platform should host alumni-specific events (e.g., an "Alumni Success Webinar Series" featuring those who became top earners in the gig economy).

- Continuous Learning and Upskilling:
    - Provide pathways for alumni to continue learning new skills. The platform could periodically offer advanced short courses or links to external MOOCs in emerging

skill areas (e.g. AI, advanced programming, etc.).

- Job/Gig Alerts:
    - Allow alumni to opt-in to ongoing gig alerts or job newsletters.
    - The platform will send out curated opportunities (from partners or scraped from gig sites) via email or WhatsApp periodically to alumni.

- Feedback and Program Improvement:
    - An alumni feedback feature where graduates can report back on how well the training prepared them for real gigs, and suggest improvements.
    - This can be as simple as a survey form, but it's an important feedback loop. Over the 5-year program, this data helps refine course content and support services. It also helps in reporting outcomes to stakeholders – success stories can be collected through the platform for publicity (with consent).

- Referral System:
    - Alumni could earn rewards for referring new applicants (since the program runs for multiple cohorts).
    - The platform can generate referral codes or links for alumni to share.
    - If someone they refer applies and gets in, the system can note that and potentially reward the alum (even if just a recognition badge or a thank-you message). This can help boost outreach for the 10 million targeted in awareness campaigns.

Platform-Wide Considerations and Architecture

Mobile Accessibility and User Experience

- Mobile-First Design:
    - Ensure the platform is fully responsive and works on all screen sizes, especially low-end Android smartphones common in Nigeria.
    - The UI should be simple, clean, and avoid heavy graphics to load well on mobile data.
    - Critical actions (like watching a lesson or taking a quiz) should be easy on a phone.

- Mobile App or PWA:

    - In addition to a responsive website, consider a lightweight mobile app or Progressive Web App.
    - A native app (Android, possibly iOS if needed) can improve experience by enabling offline access to content.
    - For example, learners could download videos or modules when they have connectivity and study offline; the app can sync progress when back online.
    - This is valuable for those with unreliable internet or power.
    - If developing an app is costly, a PWA (which can be “installed” via the browser) could be a middle ground.

- Low-Bandwidth Features:
    - Implement options to accommodate slow internet. This includes offering audio-only streams or lower-resolution video for learners on limited data.
    - Text transcripts of videos should be provided (both for accessibility and for low bandwidth reading).
    - The platform should not be too data-heavy; for instance, avoid auto-playing videos or loading large images unnecessarily.

- User Interface in English:
    - The interface will be English-only, simplifying development (no need for multilingual support initially).
    - However, it should use clear, simple English phrasing suitable for a range of education levels.
    - Use of icons and visuals can aid understanding.
    - Ensure the design could support localization later if needed (for example, in case of future expansion to other languages or regions).

- Accessibility for Disabilities:
    - Since 5% of participants are planned to be PLWD, the platform should follow accessibility best practices (WCAG guidelines). This means screen-reader compatibility, the ability to navigate via keyboard only, captioning for videos (e.g. subtitles on course videos), and options to enlarge text or high-contrast themes. These features benefit all users and ensure no one is left behind due to a disability.

Scalability and Performance Efficiency

- Cloud Infrastructure & Auto-Scaling:
    - Use cloud services to achieve high scalability without huge upfront cost.

- The platform can be containerized (e.g. using Docker/Kubernetes) to allow instances to scale out under load.
- Auto-scaling should handle spikes such as application deadlines or new course launches. For example, the application module might scale to dozens of servers during the application window for 1.5 million users, then scale down afterwards to save cost.

- Separation of Concerns (Microservices):
  - Architect the platform in a modular way. Different components – application management, LMS, community forum, etc. – can run as separate services communicating via APIs.
  - This modular design means each piece can scale independently (e.g. the LMS may need more CPU for video streaming, while the forum might need more memory for real-time chat).
  - It also allows updating or replacing modules without affecting the whole system. For instance, if a better forum software comes along, it can be swapped if the integration boundaries are well-defined.

- Caching and CDN:
  - Employ caching at multiple levels.
    - Frequently accessed data (like course lists or user profiles) should be cached in memory to reduce database load.
    - A Content Delivery Network (CDN) should be used for static assets and media – this will distribute content globally and speed up access for all users, while also reducing load on the core servers.
    - Many open-source LMS support CDN integration and cloud storage for large files.

- Testing and Monitoring:
  - Rigorously test the platform for high loads (use load testing tools to simulate thousands of users).
  - Also set up monitoring and alerting – so if response times spike or any service is nearing capacity, the tech team is alerted to act.
  - Using open-source monitoring tools like Prometheus/Grafana can ensure the platform's health is tracked.
  - This is key for a 24/7 platform where downtime would disrupt learning for hundreds of thousands.

- Cost Optimization:
  - Design with cost-efficiency in mind.

- Leverage open-source software to avoid licensing fees, and use cloud resources wisely (e.g. spot instances or autoscaling to zero for dev/test environments when not in use).
- Storing large media (videos) in cheaper object storage (like Amazon S3 or equivalent) and streaming via cloud CDNs can be more cost-effective than serving from expensive application servers.
- The platform should also remove or archive inactive data periodically to keep the working dataset smaller and costs down.

Modular Integrations and Extensibility

- Open APIs and Integration Hooks:
  - The platform should expose APIs for key functionalities (user management, course enrollment, progress data, etc.).
  - This makes it easier to integrate third-party tools and for future developers to extend the system. For example, if down the line a new AI-based assessment tool should be plugged in, having standard APIs or LTI support in the LMS will simplify integration.
  - Likewise, ensure the platform can consume external APIs – e.g. pulling gig listings, pushing data to an analytics dashboard, or verifying certificates via blockchain if needed in future.

- Third-Party Plugins:
  - Favor solutions that support plugins or extensions.
  - Open-source LMS like Moodle and Open edX have large plugin ecosystems for things like gamification, analytics, or new question types. By using such platforms, the program can add features over time (for example, a plugin for competency-based learning or a plugin to integrate WhatsApp notifications) without a complete rebuild.
  - The community edition of these tools often has many free plugins, aligning with the budget-conscious approach.

- Loose Coupling:
  - Each module (application, LMS, forum, mentorship, etc.) should be loosely coupled. They might share a single sign-on (SSO) system so users authenticate once, but otherwise operate independently.
  - SSO can be achieved with open standards like OAuth2/OpenID Connect – perhaps using an open-source Identity Server. This allows, for example, the forum to be swapped out or the LMS to be upgraded without breaking user logins. It also

means if one component goes down, others remain functional (degrading gracefully).

- Data Integration and Single Source of Truth:
    - Ensure that there is a unified database or data warehouse that pulls key info from all modules for reporting. For instance, an admin might want to see in one place: an individual's application info, their training progress, and their post-training outcomes.
    - While the operational systems can be separate, a nightly sync or an integrated reporting database can consolidate data.
    - Using open-source ETL tools or simply well-defined database schemas will help. This extensibility in data will also make it easier to incorporate new modules (say a new survey tool or a new partner's data feed) into the program's analytics.

- Partner Integrations:
    - Beyond the gig and finance integrations already mentioned, design the system to integrate with other external services as needed. For example, if the program partners with a content provider (like Coursera or local universities), the platform could integrate via LTI or API to authenticate users into those external courses.
    - Another example: integration with a national ID system for identity verification during application (if available via API) – this could streamline KYC for loans.
    - The architecture should foresee these possibilities by keeping integration points (like RESTful API endpoints, webhooks for events such as "course completed" or "certificate issued") ready for use.

Security and Data Privacy

- Secure User Authentication:
    - Implement strong authentication measures. All user accounts (applicants, learners, mentors, admins) should use secure passwords (with complexity requirements) and ideally support two-factor authentication for admins and mentors.
    - Consider SMS or authenticator app 2FA especially for administrative functions, since a breach of admin could affect many users.
    - Use encryption (HTTPS/TLS) site-wide to protect data in transit.

- Data Privacy Compliance:
    - Handle personal data in compliance with Nigeria's data protection regulations (NDPR/NDPA) and global best practices.

- Users will provide sensitive information (contact details, including IDs for KYC), so the platform must store these encrypted at rest and ensure only authorized staff can access them.
- Clearly state privacy policies to users during sign-up, explaining how their data (including training progress and any shared outcomes) will be used.
- Since the program specifically targets vulnerable populations, privacy and informed consent are crucial.

- Access Controls and Roles:
  - Define user roles with appropriate permissions – e.g., an admin can view all data, a mentor can only view their mentees, a learner only sees their own data, etc.
  - Use role-based access control in the system to prevent data leaks. For instance, if external partners (gig platforms or FSPs) access any part of the system, restrict their view (perhaps giving them a dashboard with aggregated info rather than raw personal data).

- Protecting Community Spaces:
  - As part of security, include measures to protect users in community interactions. This may involve filters for personal identifiable information (to prevent users from doxing themselves inadvertently) or anti-scam warnings if someone tries to misuse the platform to recruit or defraud participants.
  - Community guidelines and automated moderation (as mentioned) will help keep the space safe.

- Regular Backups and Recovery:
  - Given the five-year span and critical data (learning records, certifications, etc.), set up automated backups of all databases and content. Backups should be stored securely (encrypted) and tested periodically for restoration. A disaster recovery plan should exist – e.g. if the primary servers fail, a secondary standby can take over – to minimize downtime.

- Audit and Logging:
  - Maintain detailed logs of user activities and system events. This not only helps in security (tracking any unauthorized access or changes) but also in troubleshooting issues. For instance, if a user claims a course progress was lost, logs can help identify what happened.
  - Admins should have an interface to view critical logs (especially for application submissions and financial transactions) to audit the process. This also contributes to transparency and accountability in program delivery.

Cost-Effective Technology Choices

- Open-Source First Approach:
    - Prioritize open-source software for all components to avoid costly licenses. As noted, an open-source LMS (Moodle/Open edX) is recommended, and similarly open-source solutions exist for forums (Discourse), chat (Rocket.Chat), webinars (BigBlueButton), etc.
    - These can be self-hosted and customized freely. The trade-off is needing technical expertise to maintain them, but the large community support and available documentation mitigate this. An open-source stack can be hosted on affordable cloud VM instances or even on-premise servers if provided, giving full control over costs.

- Leverage Existing Platforms (with care):
    - Where open-source solutions are not mature, consider leveraging free tiers of SaaS or existing services. For example, using YouTube (unlisted videos) for content delivery is free and offloads bandwidth costs – the platform just embeds the videos.
    - Similarly, using a service like Google Forms or ODK for initial applications could be a quick, free start (though 1.5M entries may hit limits, so this might only work in early phases or need splitting across forms).
    - Always weigh the trade-offs: free external tools might have data limits or privacy concerns, so core functionality is safer on controlled infrastructure.

- Monitoring Costs:
    - Optimize database queries and storage usage to reduce costs (for example, purge or archive data no longer needed in active systems).
    - Given the non-profit nature of the project, explore if any providers (cloud or software) offer grants or credits for such initiatives – many big tech companies have programs for educational or social impact projects.

- Training and Maintenance:
    - Handover should include training of the team, a local tech team with super users or partners on managing the open-source platform.
    - Documentation of the system architecture and code is required for handover

.