Title

INTEGRATED JOB OPPORTUNITIES PLATFORM USING USSD AND ONLINE TECHNOLOGY

Author

SESTINO JOSEPHY MDECHE

Co-Author

MR. ULEMU MPONELA



Issued October 2026 Certificate

AR2026SK4E4H



ABSTRACT

Access to employment information can be difficult for job seekers who have limited access to reliable internet connectivity, smartphones, or formal recruitment networks. These challenges are particularly relevant in settings where mobile phones are widely used but access to data services and internet-enabled devices is uneven. This paper presents the design and implementation of an Integrated Job Opportunities Platform Using USSD and Online Technology, developed to connect job seekers and employers through two complementary access channels. The online interface provides web-based functions for posting and managing vacancies and for browsing and searching job opportunities. The USSD interface offers a menu-driven alternative that can be accessed using a basic mobile phone without requiring a mobile data connection. The system was developed using Python and the Django web framework, with SQLite used for data storage. Africa's Talking sandbox was used to support development and simulation of USSD interactions. The platform is intended to centralize job information, make vacancies easier to discover, and reduce dependence on a single device or connectivity model. The development process included requirements analysis, system design, implementation, and functional testing of core workflows, including job posting, browsing, searching, and USSD menu navigation. The project report describes testing against expected system behaviour; this paper therefore reports functional outcomes qualitatively and does not claim performance measurements or population-level employment impacts. The resulting prototype demonstrates how web services and USSD can be combined to support more inclusive access to employment information in Malawi. Limitations include the need for real-world deployment, broader user evaluation, reliable and current vacancy data, and validation of service availability under production network conditions. Future work may include expanded SMS alerts, improved multilingual support, recommendation functions, employer verification, and analytics for administrators. The work contributes an applied software-engineering example of a dual-channel employment information service designed around different levels of digital access.

Keywords: USSD, job opportunities, web-based recruitment, digital inclusion, Django, Malawi.

INTRODUCTION

Background of the Study

Employment information is valuable only when people can find it, understand it, and act on it in time. Job seekers may learn about vacancies through personal networks, noticeboards, social media, company websites, recruitment agencies, and online job portals. Each channel has advantages, but each can also exclude people who do not regularly use it. A web-only service, for example, assumes access to an internet-enabled device, data bundles, and sufficient digital confidence. A service based only on informal announcements may make it difficult to search across many employers or compare opportunities in a consistent way.

The project addressed this access problem by proposing a centralized platform that combines a conventional online interface with Unstructured Supplementary Service Data (USSD). USSD is a session-based mobile network service that presents short menu interactions on a handset. Unlike a typical web application, a USSD session does not require a smartphone browser or a mobile data connection. Its menu-driven nature can make it suitable for short tasks, such as selecting a location, choosing a job category, or viewing a brief vacancy summary.

The online component serves users who can access a web browser, while USSD provides a complementary channel for people using basic phones or facing unreliable internet access. The design principle is not that one channel replaces the other; rather, each channel provides a different route to a shared set of employment information.

PROBLEM STATEMENT

The central problem addressed by the project is fragmented and uneven access to job vacancy information. Job seekers may need to consult several sources, while employers may find it difficult to reach potential candidates beyond their immediate networks. Users who depend on basic phones or cannot consistently afford internet data may face additional barriers when recruitment information is distributed primarily through websites or social platforms. A centralized system with both web and USSD access offers a practical way to make job information available through more than one technology channel.

AIM OF THE STUDY

The aim was to design and implement an integrated job opportunities platform that connects job seekers and employers through web-based services and USSD access.

RESEARCH OBJECTIVES

- To identify the core information and functions needed by job seekers, employers, and administrators.
- To design a centralized platform that supports online job management and a USSD-based access path.
- To implement web functions for managing vacancies and finding job opportunities.
- To implement a menu-based USSD flow for retrieving job information through a basic mobile phone.
- To test the main user workflows against the system requirements and document limitations and possible improvements.

RESEARCH QUESTIONS

- How can web technology and USSD be integrated to provide alternative access to job vacancy information?
- What core functions are required to support job seekers, employers, and platform administrators?
- How can functional testing be used to assess whether the prototype behaves according to its requirements?
- What limitations should be addressed before the platform is deployed for wider public use?

SIGNIFICANCE OF THE STUDY

The project is significant as an applied software-engineering response to a digital-inclusion problem. It demonstrates a design in which users are not required to rely on one access technology. Job seekers with internet access can use the website, while users with basic phones can navigate USSD menus. Employers gain a centralized channel for publishing opportunities, and administrators have a defined role in maintaining platform information. The system may also serve as a basis for further research into mobile-first public information services in Malawi.

SCOPE OF THE STUDY

The scope covers the design and implementation of a prototype employment information platform, including the web application, shared data storage, and USSD interaction flow tested with the Africa's Talking sandbox. The work focuses on access to vacancy information and related platform workflows. It does not establish that the prototype has been deployed nationally, that it has increased employment rates, or that it has been evaluated with a statistically representative population.

LITERATURE REVIEW

Digital Recruitment Systems

Digital recruitment systems provide online spaces where employers can publish vacancies and candidates can search for work. Common functions include vacancy descriptions, category and location filters, employer accounts, applicant profiles, and application tracking. Centralization can reduce the effort required to find listings scattered across many sources. However, the benefits of an online recruitment platform depend on access to devices, connectivity, digital skills, and the quality of the information supplied by employers.

A job portal also needs operational processes beyond the user interface. Listings should be accurate, clearly dated, and removed or marked when they are no longer active. Employer identity and vacancy legitimacy matter because job seekers may be exposed to misleading or fraudulent advertisements. Administrative controls, reporting mechanisms, and future verification processes can therefore be important components of a trustworthy service.

USSD as an Access Channel

USSD enables short, interactive sessions over mobile networks. A user typically enters a service code and navigates text menus by replying with a number or short response. The session model is

well suited to simple information retrieval and structured choices, but it imposes constraints: messages must be concise, menu depth should be limited, and the interaction should be understandable without a graphical interface. Network timeouts and session limits also require careful design.

For an employment platform, USSD can provide a route to browse broad categories, select a district or location, and retrieve a concise description of a vacancy. It is less suitable for displaying long job descriptions, uploading CVs, or completing complex application forms. The dual-channel design therefore assigns richer interactions to the website and short information retrieval tasks to USSD.

Digital Inclusion and ICT for Development

Digital inclusion concerns whether people can access and use digital services in ways that are meaningful to them. Access is shaped not only by network coverage but also by device affordability, data costs, language, accessibility, confidence, and the relevance of the service. A multi-channel design can reduce some technology barriers, but it cannot remove every structural barrier to employment. It should be understood as an access improvement rather than a complete solution to unemployment.

In ICT for Development (ICT4D), technology interventions are commonly assessed in relation to local context, usability, institutional support, and sustainability. For this platform, relevant questions include whether users can understand the USSD menus, whether vacancy information is kept current, whether employers will maintain their listings, and who will support the service after the prototype stage.

Web Application Frameworks

Django is a Python web framework that provides routing, request handling, templates, database integration, authentication features, and an organised application structure. Its model-view-template pattern can help separate data definitions, request logic, and presentation. A framework does not automatically guarantee security or quality; developers must still configure permissions, validate input, manage secrets, protect forms, and test access-control rules.

Python was used as the main programming language, and SQLite was selected for data persistence in the project prototype. SQLite is convenient for local development and small deployments because it is file-based and requires little administration. If the platform grows to support many concurrent users or production workloads, a server-based database may be more appropriate after workload assessment.

Related Technology and Service Design

The platform combines web requests and USSD sessions with a shared application backend. Africa's Talking sandbox provides a development environment for testing USSD flows without claiming a live production service. A sandbox is useful for prototyping menu behaviour, but production deployment would require approved service configuration, a live provider arrangement, testing on target networks, and operational monitoring.

Design Principles Derived from the Review

- Provide alternative access channels while keeping the underlying vacancy information consistent.
- Keep USSD interactions short, predictable, and suitable for small screens.
- Make job listings searchable by useful criteria such as location and category where supported.
- Separate employer, job-seeker, and administrator responsibilities.
- Treat data quality, security, and user trust as ongoing operational requirements.
- Evaluate usability with real users before making broad claims about inclusion or impact.

Research Gap and Project Positioning

The project is positioned as a practical prototype that brings web-based job management and USSD access into one employment information service. Its contribution is the integration approach and the implementation of core workflows, rather than a claim to have invented web recruitment or USSD. A full empirical comparison with existing job portals or a

controlled study of employment outcomes was outside the scope of the reported work.

RESEARCH METHODOLOGY

Development Approach

The project followed a software-development approach involving problem identification, requirements analysis, system design, implementation, and functional testing. This approach was appropriate because the intended output was a working software prototype. The development process translated the access problem into user roles, workflows, data requirements, and a pair of interfaces connected to a central application backend.

Requirements Gathering and Analysis

Requirements were organised around three main user groups: job seekers, employers, and administrators. Job seekers require a way to discover vacancies and retrieve relevant information. Employers require a way to create and manage vacancy listings. Administrators require tools or processes for overseeing platform information and maintaining the service. The USSD channel adds requirements for concise prompts, numbered choices, input validation, and session continuity.

Functional Requirements

- Employers can create and manage job vacancy information through the online interface.
- Job seekers can browse and search job opportunities using the functions implemented in the prototype.
- The USSD interface presents a menu through which users can navigate job-related information.
- The application stores platform data in a centralized database.
- Administrative functions support management of system information.
- The system provides clear responses when a requested category or vacancy is unavailable.

Non-Functional Requirements

- **Usability:** web pages and USSD prompts should be understandable to their intended users.
- **Accessibility:** the system should provide a non-browser access route through USSD.
- **Reliability:** valid requests should produce consistent responses, and errors should be handled clearly.
- **Security:** user input, authentication, and role-based permissions should be handled appropriately.
- **Maintainability:** application logic should be organised so that future changes can be implemented.
- **Data integrity:** vacancy records should be stored and retrieved consistently.

System Modelling

System modelling identifies actors, processes, data stores, and the boundaries between components. The main actors are job seekers, employers, and administrators. The principal processes are vacancy management, job discovery, USSD request handling, and administrative management. The central database stores the information required by the web and USSD interfaces. The model supports a clear separation between the user-facing channels and the application logic that processes requests.

Implementation Environment

The project used Python and the Django framework for server-side application logic, SQLite for database storage, and web technologies for the online interface. USSD interaction was developed and tested through the Africa's Talking sandbox. These tools were selected to support rapid prototyping and integration using a common backend. The precise deployment configuration, hosting capacity, and production network characteristics require separate verification before operational release.

Testing Strategy

Functional testing was used to check whether

key workflows behaved according to their requirements. The test focus included job posting and management, job browsing and search, USSD menu navigation, and handling of cases where no active jobs were available. Testing in a sandbox validates the implemented interaction logic in that environment; it does not by itself demonstrate performance on every mobile network or prove that users in the field can complete tasks successfully.

Evidence and Reporting of Results

The paper reports qualitative functional outcomes based on the project report. No numerical sample size, response-time benchmark, user satisfaction score, or employment-impact statistic is claimed here because such measurements were not established in the source material. A future evaluation should define test cases in advance, record observed outcomes, and document the test environment so that results can be reproduced.

Ethical and Practical Considerations

A live employment platform may process names, contact details, CVs, application records, and employer information. Before deployment, the system should minimise collection of personal data, explain how information will be used, restrict access to authorised roles, protect stored files, and establish retention and deletion practices. Vacancy information should be checked for legitimacy, and users should be able to report suspicious listings. These are recommended deployment safeguards, not claims that every safeguard has already been implemented.

SYSTEM ANALYSIS AND DESIGN

Overview of the Architecture

The platform uses a multi-channel architecture. Job seekers and employers access the online interface through a browser. USSD users access the system through a mobile network menu. Both channels communicate with application logic responsible for interpreting requests and retrieving or updating data. SQLite acts as the central data store for the prototype. The architecture can be described as: user interface channel, application backend, database, with responses returned through the originating channel.

Web Channel

The web channel supports interactions that benefit from a graphical interface and larger amounts of information. Employers can submit vacancy details using online forms, and job seekers can inspect listings and use available search functions. The browser interface can present more detailed vacancy descriptions than a USSD screen. Web forms also require validation so that missing or malformed fields do not create incomplete records.

USSD Channel

The USSD channel is designed around short, numbered menus. The prototype menu includes options such as browsing jobs, searching by location, searching by category, accessing the user's account, and changing language. An interaction begins when the user opens the service code, then moves through menu choices. The application interprets the submitted menu path and responds with the next prompt or the requested information.

A representative menu path allows a user to choose a job-discovery option, select a location or category, and receive a concise result. When no active vacancy matches the request, the system can return a message explaining that no jobs are currently available. The project also includes English and Chichewa message examples, reflecting the importance of language accessibility. Actual language coverage and translation quality should be reviewed with fluent speakers.

Shared Backend and Data Store

A shared backend reduces the risk of maintaining separate vacancy lists for the web and USSD channels. When both interfaces retrieve information from the same source, a vacancy published through the web application can potentially become discoverable through USSD according to the system's query logic. Consistency still depends on correct database relationships, validation, and handling of active or expired records.

User Roles

- **Job Seeker**

The job seeker is the main consumer of vacancy information. The user should be able to browse

available opportunities, search using the criteria provided by the platform, and understand the information required to pursue a vacancy. The web interface can support richer information; USSD supports short discovery tasks for users who cannot use the web interface.

- **Employer**

The employer supplies vacancy information and manages listings. Useful vacancy data typically includes a title, description, location, requirements, and application instructions. The quality of results depends on employers entering accurate information and updating listings when a position is filled or closed. Employer verification is identified as a possible future enhancement where it is not already part of the implemented prototype.

- **Administrator**

The administrator supports platform oversight, including the management of system information and categories where those functions are available. Administrative responsibilities may also include handling reports of suspicious listings, resolving data-quality problems, and monitoring service health. These operational responsibilities become more important as a prototype moves toward production.

Data Flow

A typical employer data flow begins when the employer submits vacancy information through the web interface. The backend validates the request and stores the record in the database. A job seeker later sends a search request through the website or USSD menu. The backend applies the relevant selection criteria, retrieves matching records, and returns the results through the same channel. Administrative operations act on platform information under controlled permissions.

USSD Session Design

USSD sessions should be designed as short state transitions. Each prompt needs a clear question, a limited set of options, and a predictable next step. The implementation should validate unexpected input and offer a way to return to a previous menu or restart the session where possible. Since sessions can expire, the system should avoid relying on long periods of inactivity or requiring users to remember complicated sequences.

Error and Empty-Result Handling

Empty-result handling is particularly important for job discovery because the number of vacancies varies over time. A clear message such as 'No active jobs are currently available for this selection' is more useful than a blank response. Similar care is needed for invalid menu choices, missing search criteria, expired sessions, and failed connections to external services. Error messages should guide the user without exposing internal technical details.

Security Design Considerations

The system should enforce authentication and authorization on employer and administrative actions, validate all submitted data, protect state-changing web requests against cross-site request forgery, and avoid exposing sensitive application details in errors. Uploaded CVs, if accepted, require file type and size checks, safe storage, and access restrictions. Secrets and provider credentials should not be committed to source control. These controls should be verified through a security review before public deployment.

Architecture Trade-Offs

The dual-channel approach improves choice but increases design and maintenance work. Web pages can display rich information but require browser access and connectivity. USSD is more broadly compatible with basic handsets but has short message limits, session constraints, and a less expressive interface. A shared backend reduces duplication but must be designed so that each channel receives information in a suitable format. The prototype favours a practical balance between accessibility and implementation complexity.

SYSTEM IMPLEMENTATION

Application Backend

The backend coordinates requests from the web interface and the USSD handler. In a Django application, URL routing directs requests to the

relevant view or handler, application logic processes the request, and models provide an interface to stored records. This structure separates request handling from data persistence and helps organise future extensions.

Database and Vacancy Records

SQLite was used as the prototype database. A centralized database is important because the platform's value depends on consistent access to vacancy information. Data modelling should distinguish users, employer details, vacancy records, categories, and other related information as required by the implemented system. Appropriate constraints and validation help prevent incomplete or inconsistent records.

The database should support identifying active vacancies and applying search filters. A production design should also define how listings expire, how duplicate vacancies are handled, and how updates are audited. These operational policies are as important as the database technology itself.

Employer Workflow

The employer workflow begins with access to the employer-facing web functionality. The employer enters the vacancy details and submits the form. The application should validate required fields and persist valid records. The employer then needs a way to review or manage their own postings. Correct access control is important so that one employer cannot modify another employer's records.

Job-Seeker Workflow

The job-seeker workflow centres on finding useful vacancies. A user opens the web platform, browses the available jobs, and narrows the results using supported criteria such as location or category. The user then reviews the vacancy details and follows the listed application instructions. Where application submission is supported by the implementation, it should be tested as a separate workflow from merely viewing a vacancy.

USSD Workflow

The USSD workflow starts when the user enters the service code in a compatible mobile interface or simulator. The user selects from the menu, provides additional choices where required, and receives a response. The backend interprets the session data and returns a menu or result. The sandbox allows the developer to test different inputs and verify that the system responds with the expected next step.

Language Support

The project includes English and Chichewa message examples for USSD interaction. Language support is particularly relevant when users have different levels of comfort with technical or formal English. Good localisation requires more than literal translation: menu terms must be concise, familiar, and consistent, and the meaning of instructions should be checked with intended users.

Integration with Africa’s Talking Sandbox

The Africa’s Talking sandbox provides a controlled environment for prototyping USSD interactions. It allows developers to simulate session requests and examine the responses returned by the application. Sandbox testing is useful for debugging menu logic and response formatting. It should not be confused with a production deployment, which requires the appropriate service configuration, provider approvals, and validation on target networks.

User Feedback and Notifications

A job platform can benefit from timely notifications when vacancies are posted or updated. The wider project concept identifies notifications as an area for improvement and potential integration, including SMS alerts. Notifications should be treated carefully: users should control their preferences, messages should avoid disclosing unnecessary personal information, and delivery failures should be logged where appropriate. Only notification functions actually implemented and tested should be described as completed features.

Maintainability and Extension

The use of a structured web framework provides a foundation for extending the prototype. Possible extensions include a production database, employer verification, richer search, job alerts, administrative reports,

and improved localisation. Each extension should be introduced through explicit requirements, a design update, implementation, and regression testing to ensure that existing web and USSD workflows continue to work.

Deployment Considerations

A production service would require a stable hosting environment, HTTPS, secure configuration, database backups, logging, monitoring, access controls, and a documented recovery process. It would also need a process for managing vacancies and handling user enquiries. Because USSD depends on provider and network arrangements, the operational plan should include service availability, session costs, provider support, and testing on relevant mobile networks.

TESTING AND RESULTS

Testing Objectives

The purpose of testing was to determine whether the core system functions behaved according to the specified requirements. Testing focused on workflows that represent the central value of the platform: entering vacancy data, finding vacancies, navigating USSD menus, and receiving understandable responses when a search produces no results.

Interpreting the Results

The project report describes functional testing of the implemented workflows. The evidence supports reporting that the prototype was assessed against its expected functions, but it does not support numerical claims about throughput, response time, user satisfaction, or employment outcomes. For a publication, the distinction between functional correctness and real-world effectiveness is important. A menu can return the expected response in a sandbox while still requiring additional work to establish usability, reliability, and service performance in daily use.

Test Evidence and Reproducibility

A stronger future test report should include a test identifier, preconditions, input data, expected result, observed result, pass/fail status, date, software version, and environment. For USSD tests, it should also record the menu path and the complete response text. For web tests, it should identify the user role, browser, and relevant data state. Screenshots or logs can support the evidence when they do not expose private user information.

Performance Testing

Performance testing was not quantified in the material used for this paper. Before deployment, the system should be tested under representative loads to measure page response time, database query performance, USSD response delay, and concurrent session behaviour. Results should include the test environment, workload, number of requests, and summary statistics. It would be inappropriate to infer these measurements from functional tests.

Usability Evaluation

The system would benefit from task-based usability testing with job seekers and employers, including participants who use basic phones. Relevant tasks include locating a vacancy, changing a search selection, understanding an empty-result message, and posting a vacancy. Observers should record task completion, errors, time where appropriate, and participant feedback. The study should use consent procedures and avoid collecting unnecessary personal information.

Reliability and Security Testing

Reliability tests should examine network interruptions, expired USSD sessions, malformed inputs, missing database records, and external service failures. Security tests should review authentication, permissions, CSRF protection, input validation, file handling, secrets management, and error disclosure. These are recommended tests for a more complete evaluation and should not be represented as completed unless evidence exists.

Summary of Findings

The implemented prototype demonstrates the feasibility of connecting a web-based employment information service with a USSD menu flow using a common application

backend. The key result is an architectural and functional demonstration of multi-channel access. The evidence does not establish the scale of user benefit, the rate of job placement, or production readiness. These require deployment and evaluation beyond the current prototype.

DISCUSSION

Contribution of the Dual-Channel Approach

The central design contribution is the use of complementary access channels for the same employment information service. The website can present detailed listings and support richer employer interactions, while USSD provides a shorter path to selected information. This approach recognises that users have different devices and connectivity conditions. It also highlights that accessibility is a system-level property: a platform is not inclusive simply because it has a mobile interface; it must support the tasks that users need to complete.

Benefits for Job Seekers

A centralized listing service can reduce the effort of checking multiple information sources. Search functions can help users focus on opportunities that match their location or interests. USSD may extend basic access to users who do not use smartphones or mobile data regularly. These are plausible benefits of the design, but their magnitude should be assessed through user research and usage evidence rather than assumed.

Benefits for Employers

Employers can use a centralized platform to publish vacancies in a consistent format. If the service attracts job seekers through both web and USSD channels, employers may reach a broader audience than through a single online channel. However, employer value depends on the quality and relevance of candidates, the cost and effort of posting vacancies, and trust in the platform. Employer interviews and listing statistics would help assess these factors.

Digital Inclusion and Language

USSD reduces the need for an internet-enabled device but does not remove every barrier. Users still need mobile network coverage, a compatible service, and the ability to navigate text menus. Language choices and clear wording are also important. The inclusion objective should therefore be evaluated in terms of whether people with different devices, languages, locations, and levels of digital confidence can complete useful tasks.

Centralisation and Data Quality

Centralising vacancy information makes it easier to retrieve records consistently across channels, but it also concentrates responsibility for data quality. Expired, duplicated, or misleading listings can reduce user trust. A production platform should define listing expiry rules, require meaningful vacancy details, provide reporting mechanisms, and establish clear responsibilities for moderating content.

Trust and Safety

Employment platforms can be misused to advertise fraudulent opportunities or collect personal information under false pretences. The system should provide clear reporting and moderation processes, protect user data, and consider employer verification. Where a user is directed to an external application process, the platform should make that destination and any associated requirements clear.

Sustainability

A successful deployment requires more than software implementation. The service needs ongoing hosting, database maintenance, security updates, provider relationships for USSD, user support, and a reliable supply of vacancy listings. A sustainability plan should identify the organisation responsible for operating the service, how costs will be covered, and how the system will respond to policy or technology changes.

Implications for ICT4D

The project illustrates a practical principle for ICT4D: digital services should be designed around the capabilities and constraints of intended users rather than assuming universal access to advanced devices. At the same time, a technology intervention should not overstate its

social impact. The platform can improve the channels through which job information is accessed, but it cannot by itself create vacancies or resolve the broader causes of unemployment.

LIMITATIONS OF THE CURRENT STUDY

- The project is a prototype-oriented implementation rather than evidence of a fully deployed national service.
- The source material does not provide quantitative performance benchmarks or a statistically described user study.
- Testing with a sandbox cannot establish reliability on all production mobile networks.
- The availability and freshness of vacancies depend on employer participation and data maintenance.
- The USSD interface necessarily limits the amount of detail and complexity that can be presented in one session.
- Long-term sustainability, operational costs, and employment outcomes remain to be evaluated.

CONCLUSION

This paper presented an Integrated Job Opportunities Platform Using USSD and Online Technology, a prototype designed to connect job seekers and employers through a web application and a USSD-based information channel. The system uses Python and Django for application development, SQLite for data storage, and the Africa's Talking sandbox for development and simulation of USSD interactions. Its main design idea is to provide alternative routes to a common set of job vacancy information rather than require every user to access the service in the same way.

The project demonstrates how web-based vacancy management and menu-driven mobile access can be brought together within one application architecture. The online channel supports richer information and employer interactions, while USSD provides a concise alternative for users with compatible basic phones and no mobile data connection. Functional testing of core workflows provides a basis for assessing whether the prototype behaves according to its intended requirements.

The study does not claim that the prototype has increased employment rates, achieved national-scale access, or met quantified performance targets. Those questions require deployment, broader user evaluation, and carefully designed impact assessment. Further work should focus on security and privacy, vacancy quality, employer verification, production network testing, language accessibility, notifications, and sustainable operation. With these measures, the platform could provide a useful foundation for more inclusive digital employment information services in Malawi.

Acknowledgements

The author acknowledges the academic guidance and project support received during the development of the system. Please confirm the supervisor's name, institutional wording, and any required acknowledgements before submission.

Author Contribution

The author is listed as the project author. If other people contributed substantially to the research, software development, supervision, or writing, their roles and authorship status should be agreed before publication.

Conflict of Interest

The author should declare any actual conflict of interest according to the target publication's requirements. Do not submit this statement unchanged if a relevant conflict exists.

Data Availability

No separate research dataset is described in the source material used for this manuscript. If code, test records, or other materials are to be shared, add a verified repository link or describe the applicable access conditions.

REFERENCES

1. Africa's Talking. (n.d.). USSD documentation and developer resources. Verify the exact documentation title and URL used during implementation before submission.

2. Django Software Foundation. (n.d.). Django documentation. https://docs.djangoproject.com/ (Verify the version and access date relevant to the project.)

3. GSMA. (n.d.). Mobile technology and digital inclusion resources. Verify the exact report or publication cited and its publication details before submission.

4. International Labour Organization. (n.d.). Employment and labour market resources. Verify the exact report, title, year, and URL relevant to the discussion.

5. World Bank. (n.d.). Digital development and employment resources. Verify the exact publication and bibliographic details before submission.

6. United Nations Development Programme. (n.d.). Digital inclusion and development resources.