Cureus Journal of Computer Science

Open Access Original Article

Revitalising the Irula Language Through Mobile Language Learning Application

Arul Dayanand [1] , Uma Devi [2] , Kevin Thomas Jinu [3] , Mohammed Ariz Siddiqui [4] , Evina Novi Thekkudan [5]

1. Dr. T. R. Paarivendhar Multidisciplinary Research Centre for Indigenous Language Studies, Revitalization, and Computational Resources, Department of English and Foreign Languages, SRM Institute of Science and Technology, Kattankulathur Campus, Chengalpattu, IND 2. Department of Computing Technology, SRM Institute of Science and Technology, Kattankulathur Campus, Chengalpattu, IND 3. School of Computer Science, University of Sydney, Sydney, AUS 4. Department of Computer Science, University of Texas, Dallas, USA 5. Department of Computer Science and Engineering, University of Minnesota, Minnesota, USA

Corresponding author: Arul Dayanand, aruldays@srmist.edu.in

Received 06/30/2025
Review began 08/22/2025
Review ended 11/12/2025
Published 11/13/2025

DOI:
https://doi.org/10.7759/s44389-025-08635-7

Abstract

Language serves as a fundamental conduit for transmitting culture within Indigenous communities. The cessation of language use poses a significant risk of extinction, consequently leading to the loss of substantial historical knowledge. India, characterised by its multicultural and multilingual landscape, has 449 indigenous tribal communities. In Tamil Nadu, there are approximately 36 scheduled tribes, six of which are classified as primitive tribal groups. The Irula community is one such group, whose native language is endangered due to factors such as language shift, discrimination, undervaluation of the language, and, notably, the younger generation's lack of interest and motivation to learn it. Mobile-assisted language learning (MALL) technologies have facilitated the development of mobile applications, including games, dictionaries, and translators, which support autonomous learning and revitalisation of languages. This study aims to design and develop an Android-based application that utilises MALL to engage young language learners within the Irula community. This study employs secondary data sources to construct a linguistic dataset, while agile software development methodologies are applied in the creation of MALL. This article elaborates on the architecture and design of MALL. Experimental evaluation on six devices demonstrated stable performance across both iOS and Android platforms, with positive initial user feedback on usability, confirming the app's potential as a tool for Irula language revitalisation.

Categories: Software Development, User Interface Design (UID), Digital Humanities
Keywords: irula, language revitalization, mobile app, indigenous communities, mobile-assisted language learning

Introduction

Language endangerment is a significant global issue, with research suggesting that approximately 46% of the world's 7,000 language communities are at risk of losing language transmission entirely by the end of this century [1]. This trend threatens cultural diversity and undermines the social cohesion of Indigenous and minority groups worldwide. Against this global backdrop, this study focuses on the plight of the Irula language in India, a poignant example of linguistic vulnerability.

India hosts 462 individual languages, 197 of which are classified as endangered by UNESCO's Atlas of the World's Languages in Danger, encompassing categories from "vulnerable" to "extinct" [2]. Among these languages, Irula - spoken primarily by a community of 189,661 individuals in Tamil Nadu (with 9,480 residing in the Nilgiris) and by smaller populations in Kerala and Karnataka - faces rapid attrition as speakers shift to dominant regional tongues such as Tamil, Malayalam, and Kannada [3]. This shift accelerates the erosion of cultural heritage and weakens the community bonds.

Traditional preservation efforts for the Irula language, such as printed dictionaries, classroom instruction, and archival recordings, have made important contributions but remain constrained by limited accessibility, dwindling numbers of fluent speakers, and challenges in sustaining learner engagement. Therefore, there is an urgent need for innovative, scalable approaches that can overcome these barriers and facilitate continuous and autonomous learning.

Mobile-Assisted Language Learning (MALL) offers a compelling alternative. MALL refers to the use of mobile devices, such as smartphones and tablets, to support language learning through interactive, on-demand applications [4]. By leveraging the ubiquitous nature of mobile technology, MALL can deliver customised content, real-time feedback, and multimedia resources directly to learners, including those in resource-limited settings. Compared to conventional methods, MALL enhances learner motivation, encourages self-directed study, and bridges geographic and infrastructural gaps [5]. Studies in India and other contexts have demonstrated MALL's effectiveness in improving language proficiency among remote and underserved populations while highlighting challenges such as connectivity constraints and device availability [6].

How to cite this article
Dayanand A, Devi U, Jinu K, et al. (November 13, 2025) Revitalising the Irula Language Through Mobile Language Learning Application. Cureus J Comput Sci 2 : es44389-025-08635-7. DOI https://doi.org/10.7759/s44389-025-08635-7

In this study, we employed a mixed-methods approach to evaluate the feasibility of MALL for the revitalisation of the Irula language. This process commenced with secondary collection of lexical and cultural data from language archives. These data will guide the design and development of a multilingual mobile application using an Agile framework. Finally, the mobile application was installed on multiple devices to evaluate its performance. This study distinguished itself by focusing on the specific sociolinguistic context of the Irula community, employing a cross-platform mobile development framework (React Native) for wider accessibility, and integrating a bilingual (English-Tamil) interface to bridge the language shift gap between generations.

Problem statement

The younger generations within the Irula community are increasingly growing up without proficiency in their native language as they adopt more dominant languages such as Tamil, Malayalam, and Kannada. This presents a significant challenge for the Irula community, as it poses a critical threat to the survival of their language. The erosion of the Irula language endangers the community's unique cultural heritage and the traditional knowledge transmitted across generations, potentially leading to a breakdown in social cohesion. To address this issue, the study's objective is to revitalise the Irula language by developing a mobile application for language learning that will facilitate straightforward language acquisition for younger individuals. This application aims to provide an effective means of preserving and revitalising the Irula language while fostering a sense of pride among the Irula people in their cultural heritage and longstanding traditions.

Purpose of the study

This study aimed to develop a mobile language-learning application that functions as an accessible Irula-language dictionary. The primary aim is to preserve and revitalise the endangered Irula language while raising awareness of the significance of preserving indigenous languages globally. Given the scarcity of available written materials for learning Irula, the mobile application offers Irula speakers, particularly the younger generation, a readily accessible resource for learning and understanding the language. It incorporates features such as audio pronunciation, example sentences, and explanations of grammatical rules. Furthermore, the application can serve as an archive for linguists and researchers interested in examining the unique characteristics of the Irula language, thereby contributing to a broader understanding of language.

Literature review

Comprehensive Approach to Language Learning and Maintenance Through Technology

Technology is increasingly being employed in language acquisition and preservation for minority groups. In one study, an Android dictionary application was developed for the Lampung language [7], containing 15,000 words in Lampung and Indonesian with audio recordings by native speakers. This application helps raise awareness of the Lampung language among younger generations. A mobile application was developed in Papua, Indonesia, for the indigenous Muyu language [8]. Although Muyu students found the application adequate, further research is needed to assess its effectiveness as a learning medium. An Android-based medium for Arabic language learning was developed to enhance student interest [9]. Although deemed feasible for education, it lacked usage instructions and required material design improvements. An Android application was created for elementary school children to teach the Mongondow language, covering recognition, numbers, colours, and animals [10]. While mobile learning was facilitated, the application was limited to Android. These studies demonstrate the potential of technology for language acquisition and maintenance in minority populations. Mobile applications provide access to linguistic resources and interactive learning. However, research is needed to evaluate their effectiveness and address the implementation challenges in communities with limited technological access.

Exploring the Potential of Mobile-Assisted Language Learning and Other Tools in Language Education

The integration of technology in language education has notably increased, with mobile applications becoming a prevalent tool for teaching English to specialised groups. The study conducted by Charif [11] assessed the efficacy of mobile applications for teaching legal English to Algerian law students, concluding that these applications offer flexibility, convenience, and personalised learning experiences. However, this study is constrained by a small sample size and the absence of pronunciation features. Research on MALL in English for Specific Purposes [12] indicates that it facilitates flexible and accessible learning, although it is similarly limited by sample size and study duration. A case study [13] on the use of Flutter and Firebase illustrates the development of cross-platform e-learning applications accessible via mobile and web browsers, although it lacks support for desktop platforms. These publications underscore the role of technology in language and e-learning, demonstrating the advantages of adaptability, convenience, and individualised learning while also acknowledging limitations, such as small sample sizes and missing features. Further research is necessary to fully comprehend the potential of these technologies across various contexts and enhance their effectiveness in language education. These

insights are valuable for educators and developers seeking to integrate technology into their teaching methodologies.

Materials And Methods

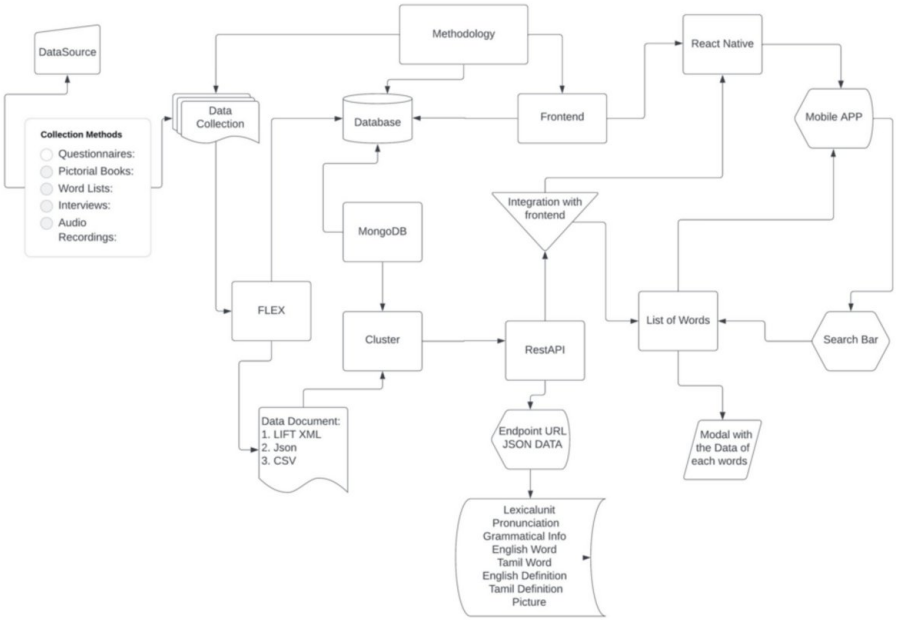

FIGURE 1: UML Diagram

UML, Unified Modelling Language

Figure *1* delineates a methodology for the collection and storage of linguistic data, comprising three principal components: data collection, databases, and front-end. During the data collection phase, various secondary sources, including surveys, pictorial books, and audio recordings, were employed to gather data. The collected data were documented using a fieldwork tool and subsequently converted into LIFTXML, CSV, and JSON code formats. The database component establishes a cluster for the storage of JSON-formatted data using MongoDB. A REST API was developed using a JSON format file that encompasses data elements such as lexical units, pronunciation, grammatical information, English and Tamil words, definitions, and images. This REST API was integrated with the front end via React Native, facilitating the creation of a mobile application. Within the mobile application, the list of words is presented as a flat list; selecting an item from the list opens a modal display of additional JSON data from the REST API. Furthermore, a search bar was implemented to optimise word searches. Figure *2* illustrates how the architectural diagram provides a comprehensive strategy for collecting, storing, and retrieving linguistic data through the application of modern technology.

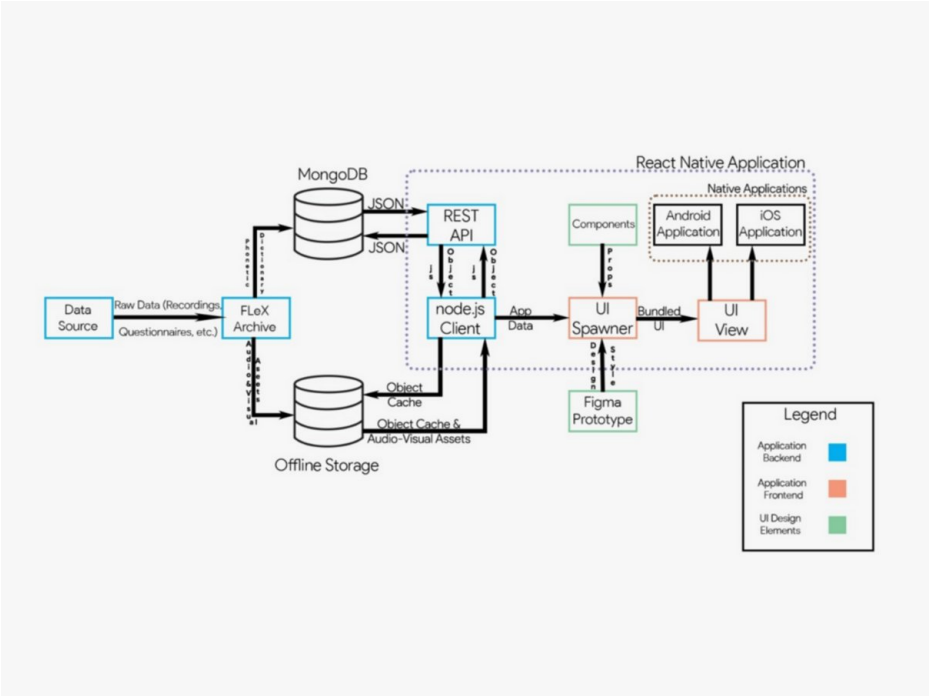

FIGURE 2: Mobile-Assisted Language Learning Architecture

API, Application Programming Interface; UI, User Interface

Data collection

In the initial experimental framework of this study, linguistic data pertaining to the Irula community were gathered from secondary sources using various methods, including books, journals, articles, pictorial books, word lists, and digital archives. The collected data will include the phonetics of Irula words, along with translations, definitions, examples, images, and audio files in Tamil and English. ELAN or Praat software will be used to segment and annotate the data, and all data will be stored in a Lexicon Interchange Format (LIFT) file generated by the FLEx tool. This file was subsequently converted to develop our backend application programming interface (API).

Backend development

The project involves backend development in Figure *3*, which encompasses the conversion of data from the LIFT format to XML and JSON to facilitate expedited processing. It also includes the storage of JSON data files, audio files, and image files within a MongoDB cluster and the development of a RESTful API to provide data access to the front end. The API interacts with the front-end user interface, transmitting data periodically and permitting manual updates to the dictionary. A dual storage strategy, which retains all assets and dictionaries on the device for offline accessibility, was devised considering the application's environment and utility. MongoDB is employed as a NoSQL-based database system to provide storage for the application.

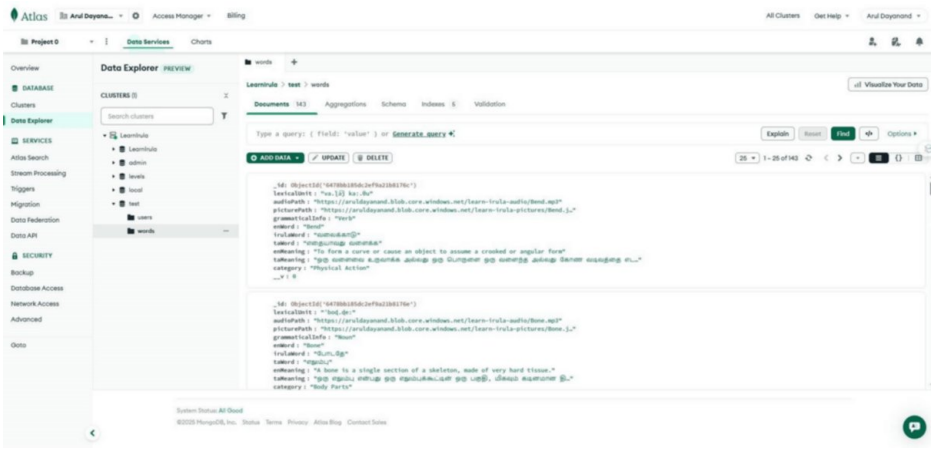

FIGURE 3: Backend Development

User interface

The User Interface (UI) was meticulously designed to prioritise simplicity and intuitiveness, resulting in a straightforward yet functional application. The app prototype was developed using Figma and adapted for both English and Tamil, as illustrated in Figure *4*. The UI employs a minimalistic design pattern to enhance functionality, complemented by a high-contrast colour palette to facilitate differentiation of various elements. Within the interface, each data element expands into a modal that contains all the pertinent information about the element.

FIGURE 4: App Home Screen

Frontend development

Mobile App

In developing the mobile application for conserving the Irula language, the team employed React Native in conjunction with Expo CLI. This approach facilitated the creation of a unified codebase that is applicable to both iOS and Android platforms. The comprehensive library of prebuilt UI components offered by Expo CLI significantly reduced the time and effort required for user interface development (Figure *5*). React Native's capability to produce efficient and high-performance applications, along with its native components and APIs, ensures optimised performance and seamless operation on mobile devices. The use of React Native with Expo CLI resulted in the creation of a high-quality, efficient, and user-friendly mobile application for the Irula community.

FIGURE 5: Overview of the App

Integration With RESTful APIs

To fetch data from the MongoDB backend and display it on the mobile application, we integrated a RESTful API using Node.js and Express. The API provides endpoints for retrieving data in the JSON format. We used the Axios library to fetch data from the API and display it in a FlatList component, which is optimised for rendering large lists of data in a mobile app. By hosting the API using MongoDB, we were able to gain greater control over the data and ensure its security. We mapped the fields in the JSON data to the fields required by the application by creating a schema for the data in the MongoDB database, defining the fields and their respective data types, and any validation rules that needed to be enforced. The API endpoints were configured to return the data in the required format, and the React Native application used the Axios library to make HTTP requests to these endpoints and retrieve the required data. Finally, we rendered a FlatList component that displayed the data. The integration of the Irula Language

Conservation Project with a RESTful API was critical for accessing the required data stored in MongoDB clusters.

Modal

When a word is clicked in the app, a modal opens that fetches the word's audio file and image file and displays them along with other linked data from the API.

Search Bar

The search bar in our mobile app is an essential component that enables users to effectively search for words. We created a case-insensitive early termination linear search algorithm to reduce the number of required iterations. We also added a debounce method to avoid making multiple API calls whenever we modified the search text. These modifications improved the functionality and usability of the search bar, making it faster and easier to use for our customers. Overall, the search bar is a helpful feature that enhances the user experience of the mobile app.

Agile development methodology

Agile software development is a methodology that prioritises adaptability and collaboration, which makes it suitable for a wide range of industries and projects. One of the primary advantages of Agile is its adaptability to changing requirements and priorities. Throughout the development process, the agile methodology incorporates check-ins and feedback from stakeholders, allowing for adjustments and modifications. This makes it suitable for projects where the requirements or objectives are not fully defined at the outset or where there is high uncertainty or risk involved.

Results And Discussion

This section delineates the findings and analyses a study conducted to assess the LearnIrula mobile application, which is designed to preserve and revitalise the endangered Irula language. The study encompassed a device testing phase, during which the application was installed on multiple smartphones to evaluate its performance.

Experimental setup

To evaluate the performance and usability of the LearnIrula mobile application, experimental testing was conducted across six devices, representing hardware specifications and operating systems. These included four Android devices and two iOS devices to ensure a comprehensive cross-platform assessment. The evaluation focused on application performance, load time, frame rate, RAM usage, and user interaction efficiency.

Hardware Environment

Android and iOS devices were selected to represent diverse processing capabilities, Random Access Memory (RAM) capacities, and screen resolutions. Android devices used were as follows: OnePlus 7T (OS 11, 8 GB RAM, 6.55″), Galaxy C9 Pro (OS 8, 6 GB RAM, 6.0″), OnePlus Nord CE 5G (OS 11, 6 GB RAM, 6.43″), and OnePlus Nord 2 (OS 12, 8 GB RAM, 6.43″). iOS devices used were as follows: iPhone 14 Pro (iOS 16.4.1, 6 GB RAM, 6.1″) and iPhone 6s (iOS 15.7, 2 GB RAM, 4.7″). This selection ensured a comprehensive evaluation across different hardware specifications and operating system versions, providing a robust testing environment for the study.

Software Environment

The application was built using React Native with Expo CLI, providing a single codebase for both iOS and Android platforms. The backend infrastructure utilises MongoDB in conjunction with a Node.js/Express REST API. For testing and optimisation purposes, the development team employed a range of tools, including the Expo performance profiler, network monitoring tools, and device-native debugging consoles. This comprehensive tech stack enables efficient cross-platform development, robust data management, and thorough performance analysis to ensure a smooth and responsive user experience across various devices and operating systems.

Test Procedure

The latest build of the LearnIrula app was installed on each test device, and the initial launch time and RAM usage were verified using performance profiler logs. Standard user interactions were conducted, including searching for words, opening modals, playing audio files, and navigating menus. API response times and search rendering performance were measured under stable Wi-Fi (100 Mbps) and limited 4G

connectivity to assess network resilience. Any functional errors, interface inconsistencies, or latency spikes encountered during testing were documented. Qualitative feedback was gathered from five Irula community members and two software testers to assess the app's usability and cultural appropriateness.

Evaluation Metrics

This study evaluated performance, usability, compatibility, and user satisfaction as the key metrics. Performance was assessed using load time, RAM usage, frame rate, and hydration time measurements. Usability focuses on search accuracy, modal loading speed, and ease of navigation. Cross-platform stability and UI consistency across different screen sizes were used to test compatibility. User satisfaction was gauged by the perceived ease of use, visual appeal, and learning value. This comprehensive approach to data collection ensured that the results obtained were reproducible, measurable, and representative of the real-world usage scenarios. Considering these diverse aspects, the experiment aimed to provide a holistic understanding of the system's effectiveness and user experience.

Device Testing

During the testing phase, the LearnIrula application was installed on multiple smartphones with various screen types, and its performance was evaluated. Table *1* presents the specifications and results of the device testing.

S. No.	Device	Operating System/Specs	Load Time (s)	Frame Rate	RAM Usage	Interface Display	Menu Functions
1	OnePlus 7T	Android 11, 8GB, Octa-core, 6.55″	1.1	60 fps	210 MB	Displayed correctly	All menus functional
2	Galaxy C9 Pro	Android 8, 6GB, Octa-core, 6.0″	1.3	58 fps	235 MB	Displayed correctly	All menus functional
3	iPhone 14 Pro	iOS 16.4.1, 6GB, Hexa-core, 6.1″	1	60 fps	205 MB	Displayed correctly	All menus functional
4	OnePlus Nord CE 5G	Android 11, 6GB, Octa-core, 6.43″	1.2	59 fps	225 MB	Displayed correctly	All menus functional
5	iPhone 6s	iOS 15.7, 2GB, Dual-core, 4.7″	1.7	56 fps	260 MB	Displayed correctly	All menus functional
6	OnePlus Nord 2	Android 12, 8GB, Octa-core, 6.43″	1	60 fps	215 MB	Displayed correctly	All menus functional

TABLE 1: Device Testing

Across all devices, the application exhibited a consistent rendering performance, sustaining an average frame rate of 59 fps and an average launch time of 1.4 seconds. The hydration cycle, encompassing the complete data fetch, execution, and rendering time, averaged 1.7 seconds, signifying efficient backend data delivery and frontend rendering. The iPhone 6s (2 GB RAM) recorded the highest load and hydration times, yet maintained full functional validity, demonstrating compatibility with low-memory devices. This outcome demonstrates the lightweight optimisation and low processing demands of the application.

Usability and User Testing

User testing was conducted with seven participants, comprising five members of the Irula community and two professional software testers. Participants engaged with dictionary entries, listened to pronunciation audio files, viewed images, and navigated the bilingual (English-Tamil) interface. The search bar's implementation of an optimised linear search algorithm, incorporating early termination and a debounce method, effectively minimised redundant API calls. The average search response time was recorded at 180 ms over Wi-Fi and 230 ms over 4G networks, indicating robust network resilience. In modal loading, each lexical entry was loaded with a median time of 1.5 seconds, which included the retrieval of associated audio and image files. Regarding the feedback, participants described the interface as "easy to use," "visually clear," and "helpful for learning pronunciation." Importantly, no critical user interface issues were reported, providing reassurance about the system's usability. The primary user's suggestion was to incorporate bookmarking and example sentences to facilitate extended learning.

Discussion

Performance and System Analysis

To evaluate various aspects of our system, we employed a range of performance metrics to ensure the efficient operation of our application. For the front end and user interface, several tools and metrics are available, contingent upon the deployment type, to assess the performance of an application integrated into the React Native Bundler. We used Expo to configure and deploy the application, which provided a convenient graphical overlay format. The primary indicators of an application's performance are its load time and RAM consumption. In both instances, lower values are preferable because they signify the efficient use of system resources.

Another method for measuring performance is the application frame rate. This is also known as the application performance rate. In terms of non-gaming applications, the frame rate refers to the highest number of screen updates that a processor can push for an application. The higher the number, the better the performance. Other beneficial tooltips available in Expo applications include ScriptDownload and ScriptExecution. These names are self-explanatory and clearly explain the metrics they signify.

Another way of measuring load times for applications is called the time taken to render, otherwise known as “hydration time”. This signifies the time taken by a full fetch, execute, and render cycle of the application. The rendering time of the API and search handle is measured. This value will vary with each "hydration cycle" and represents the average loading time from when the application opens until it is ready for user interaction, considering network connections. Again, the lower the number, the better the result.

Experimental Analysis

The evaluation of the LearnIrula application was conducted on six devices with varying hardware configurations and operating systems. This demonstrated consistent functional validity and cross-platform stability. The application maintained a responsive interface across Android (OS 8-12) and iOS (15.7-16.4.1) devices. This indicates a robust adaptability to diverse system environments. Performance metrics revealed that the average launch times and hydration cycles were within acceptable mobile application standards. The frame rates remained stable for all the tested devices. This suggests efficient rendering by the React Native and Expo frameworks, even on lower RAM configurations, such as the iPhone 6s (2 GB). The optimised search algorithm, with debounce functionality, significantly reduced the number of redundant API calls. This lowers the average search latency and improves the user experience in limited connectivity scenarios.

Usability observations indicated that the modal-based word display effectively supported audio-visual learning. The bilingual (English-Tamil) interface ensured cultural accessibility. However, feedback from the Technology Acceptance Model survey suggested the inclusion of additional features in the app. These include (Irula Language Dataset) word usage examples, synonyms and antonyms, and bookmarking to enhance learner engagement. Device compatibility analysis confirmed no functional degradation across various screen sizes and resolutions. UI consistency was maintained throughout the process. This suggests that the chosen front-end architecture and high-contrast minimalistic design are effective for maintaining accessibility standards. Overall, the experimental results validate the potential of the LearnIrula application as a scalable MALL solution for endangered language revitalisation. Performance stability, usability across devices, and user acceptance data support its deployment in real-world community learning environments. Iterative feature enhancement should address current functional gaps.

These findings corroborate previous research on indigenous language learning applications, such as those developed for Lampung [7] and Muyu [8], which reported enhancements in accessibility but encountered technical limitations. By contrast, LearnIrula offers cross-platform compatibility, offline functionality, and bilingual learning support, representing a significant advancement in mobile-based language revitalisation. By maintaining device stability with minimal hardware resources, the application provides an accessible tool for the revitalisation of endangered languages. Its consistent performance also indicates its potential for replicability across other South Dravidian tribal languages.

Conclusions

In conclusion, this study demonstrates that the developed MALL application for the Irula language exhibited stable performance across all tested Android and iOS devices. The application was favorably received by both community members and testers, who reported ease of use, cultural appropriateness, and value of audio-visual resources. Quantitative data indicated short load times and low RAM usage across devices with a consistent and functional user interface. These findings confirm the technical and practical feasibility of deploying such an application to support endangered language revitalisation within and beyond the Irula community. Future research should encompass large-scale community deployment, longitudinal studies on language retention and learning impact, and collaboration with educators to

expand features, such as bookmarking and usage examples. Further investigation is required to ensure the app's sustainability, scalability, and support in areas with limited connectivity, as well as ongoing maintenance and dataset growth.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Arul Dayanand, Uma Devi, Kevin Thomas Jinu, Mohammed Ariz Siddiqui, Evina Novi Thekkudan

Acquisition, analysis, or interpretation of data: Arul Dayanand, Uma Devi, Kevin Thomas Jinu, Mohammed Ariz Siddiqui, Evina Novi Thekkudan

Drafting of the manuscript: Arul Dayanand, Kevin Thomas Jinu, Mohammed Ariz Siddiqui, Evina Novi Thekkudan

Critical review of the manuscript for important intellectual content: Arul Dayanand, Uma Devi

Supervision: Arul Dayanand, Uma Devi

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Acknowledgements

This article was previously presented at a two-day national seminar on "Tribal Society, Their Linguistic Landscape, and Indigenous Knowledge: A Plan for Preservation and Promotion" by the Department of Tribal Studies, Art, Culture, and Folk Literature, Indira Gandhi National Tribal University (A Central University), India, on 20th–21st March 2023. The authors used the Grammarly AI tool to correct and improve the academic writing in the paper. Irula Language
Dataset: "https://drive.google.com/file/d/1mGZ51y9X3qeK6jhsSC65Jfr2W8nZgUgs/view?usp=sharing"

References

1. New Estimates on the Rate of Global Language Loss. Rosetta Project. (2013). Accessed: June 22, 2025: https://rosettaproject.org/blog/02013/mar/28/new-estimates-on-rate-of-language-loss/.
2. Leonard WY: Reframing language reclamation programmes for everybody's empowerment. Gender and Language. 2012, 6:339-367. 10.1558/genl.v6i2.339
3. Hinton L: Language revitalization: An overview. The Green Book of Language Revitalization in Practice. Hinton L, Hale K (ed): Brill, Boston; 2001. 1-18. 10.1163/9789004261723_002
4. Darwin L: Language of Irulas: A Linguistic Study [Doctoral dissertation]. University of Kerala, Thiruvananthapuram. (2007). Accessed: June 24, 2025: https://shodhganga.inflibnet.ac.in/handle/10603/402546.
5. Mobile-Assisted Language Learning. Wikipedia. (2022). Accessed: June 24, 2025: https://en.wikipedia.org/wiki/Mobile-assisted_language_learning.
6. Gangaiamaran R, Pasupathi M. Review on the use of mobile apps for language learning. (2017). Accessed: June 24, 2025: https://www.researchgate.net/publication/322465408_Review_on_use_of_mobile_apps_for_language_learning.
7. Ariyani F, Putrawan GE, Riyanda AR, Idris AR, Misliani L, Perdana R: Technology and minority language: an Android-based dictionary development for the Lampung language maintenance in Indonesia. Tapuya: Latin American Science, Technology and Society. 2022, 5:10.1080/25729861.2021.2015088
8. Tambaip B, Wayangkau IH, Suwarjono S, Loupatty M, Adam AF, Hariyanto H: Preservation of the Muyu indigenous language with an Android-based dictionary. Emerging Science Journal. 2021, 4:85-101. 10.28991/esj-2021-sp1-06
9. Mizan AN, Aridan M, Akmansyah M, Adelia S, Asiyah IN, Sholikha LM: The development of Arabic learning media based on Android for senior high school students. LISANIA: Journal of Arabic Education and Literature. 2022, 6:178-189. 10.18326/lisania.v6i2.178-189
10. Mewengkang A, Liando OES, Ngodu MR, Moningkey ERS, Wantania T: Android-based application for children learning with Indonesian and Mongondow language. Proceedings of the 5th UPI International Conference on

Technical and Vocational Education and Training. 2019, 475-478. 10.2991/ictvet-18.2019.109

11. Charif MM: Designing a course for the teaching of legal English with implementing mobile applications . Applied Linguistics. 2022, 6:104-118.
12. Khan RMI, Radzuan NRM, Alkhunaizan AS, Mustafa G, Khan I: The efficacy of MALL instruction in business English learning. International Journal of Interactive Mobile Technologies (iJIM). 2019, 13:60-73. 10.3991/ijim.v13i08.9562
13. Al-Atraqchi OMA: Backend as a service cloud computing integrated with cross-platform mobile development framework to create an e-learning application that works in mobile and web with a single codebase. 4th International Conference on Communication Engineering and Computer Science (CIC-COCOS'2022). 2022, 50-58. 10.24086/cocos2022/paper.511