ELEMENTAL

eGuide to Building a Multilingual EdTech Platform

Unifying Development and Localisation to Create EdTech Platforms Ready for the World

LANGUAGE INSPIRED

Table of Contents

01 Introduction

02 Why Localization Matters for EdTech Growth

03 Common Localization & Development Challenges

04 Best Practices for Multilingual EdTech Platforms

05 Actionable Steps to Implement Localization & Multilingual Development

06 Case Studies & Success Stories

07 Conclusion & Next Steps

Introduction 01

"The rise of multilingual EdTech platforms: Why going global is essential.

Going global allows EdTech platforms to reach diverse learners, expand their market presence, and drive greater impact by making education accessible across regions. Implementing multiple languages *ensures inclusivity, improves user engagement, and enhances learning outcomes* by delivering content in a way that resonates with local audiences.

Challenges of localization in EdTech-beyond simple translation.

There are of course many challenges when it comes to the implementation of localization in EdTech platforms, far beyond simply translating words from one language to another. These include **UI and UX adjustments**, **technical adjustments to the platform**, **content adaption and linguistic testing**. All of these aspects we will cover in this eGuide.

How this guide helps: Combining technical implementation and language expertise.

This guide will bridge the gap between technical implementation and language expertise by providing practical strategies for integrating multilingual capabilities into EdTech platforms, while ensuring cultural and pedagogical accuracy. It will offer insights on leveraging technology for seamless localization, managing linguistic nuances, and maintaining consistency across different languages *to create an effective, user-friendly learning experience*.

Who is this eGuide aimed at?

If you're a product owner, marketing manager, CTO or founder of an EdTech platform, this guide is for you!

About the authors

Elemental is a web and software development agency based in Cape Town, South Africa. The company has two decades of experience, designing and developing custom and complex web apps, software and mobile apps, with a specific focus on the EdTech Industry, amongst others. Find out more here: https://www.elemental.agency

Language Inspired is a global language solution company that uses a unique human-technology hybrid approach. Since 2009, we've helped 2,000+ businesses expand worldwide, delivering content in 200+ languages. Our native linguists bring industry-specific expertise, including Edtech, to ensure accurate, culturally relevant localization that meets the highest standards. You can find out more here: https://www.languageinspired.com/

Why Localization Matters for EdTech Growth

02

We'll get straight to the point and tell you why. When users can learn in their native language, they learn faster, stay engaged, and ultimately feel more connected to your platform. When that happens, you get a higher ROI and a better shot at scaling over competitors.

Now, let us tell you how it works.

Global Product Experience
Language is at the core of how people learn. Getting the basics right is the fundamental step to creating a cohesive experience around your product. While this includes your main platform, it must also include the touchpoints before and after **(e.g. marketing materials, customer support, etc.)** users engage with your brand and website.

Global User Adoption
Language accessibility not only attracts users to your platform but also ensures they stay. A study by the European Commission found that 9 out of 10 internet users prefer websites in their native language, even if they understand English. The same principle applies to EdTech learners are more engaged when the content feels familiar.

Global Scalability
Future-facing companies will also have to think about how they integrate localization that scales as their business grows. For instance, are you localizing after the development sprint (Waterfall model), or is localization built into the process (Continuous localization)? While the first approach results in more time-to-market, the second ensures real-time platform updates across all languages.

But don't just take our word for it. Here are some language stats that highlight loud and clear why localization matters.

-UNESCO data shows 40% of the global population does not have access to education in a language they speak or understand. However, we learn significantly better and achieve greater academic success when taught in a language we fully understand.

-52.4% of online consumers agree that they will never use a service from a site that supports only English.

-You may lose up to 8-13% of customers when you don't translate content into people's preferred languages.

-84% of businesses see a moderate to high revenue growth when they invest in localization.

The increasing ease of doing business across borders means EdTech is no longer reserved for the few. However, to truly serve this global audience, you must build a global-ready product, which means developing it in a way that it's easy to adapt to different market needs, especially when it comes to language localization.

Common Localization & Development Challenges 03

3.1 **Technical Development Challenges**

- Structuring platforms for future multilingual expansion.

If you know your EdTech platform may need to support multiple languages in the future-even if you're launching with just one-it's far more effective to build that capability in from the outset.

By planning for multilingual support from day one, your platform's technical foundation (from database architecture to application logic) can be structured ***to easily accommodate additional languages***. This forward-thinking approach helps avoid major rework later and ensures the codebase remains clean, modular, and scalable.

In contrast, retrofitting multilingual support into an existing platform can be a complex, time-consuming, and costly exercise. Many components of your platform-such as JavaScript functions, UI elements, and third-party tools-may break or behave unpredictably if they weren't initially built with localization in mind.

For example, third-party integrations like image uploaders or newsletter signup widgets often have hard-coded text or lack multilingual support altogether, requiring manual overrides or workarounds that are not always straightforward.

Moreover, implementing these changes post-launch can be even more challenging. By that stage, your platform may already have been live for months or years, with a growing user base, existing content, and dependencies that make sweeping changes riskier and more disruptive.

- The Role of Tech Stack in Multilingual Success

Your chosen tech stack plays a critical role in how smoothly your platform handles multilingual expansion. For example, while WordPress *can* support multiple languages, it's not ideal for complex implementations. Two key limitations include:

- **Plugin Reliance** – Multilingual functionality often depends on third-party plugins like WPML or Polylang, which can introduce performance issues or conflicts with other plugins.
- **Limited Flexibility** – WordPress was not inherently designed for multilingual content, which can make structured language switching and localized UX flows cumbersome to implement.

On the other hand, modern **JavaScript frameworks** like **React or Angular** are better suited for scalable, multilingual platforms. These frameworks offer powerful i18n (internationalization) libraries and built-in features that allow for efficient language switching, dynamic content loading, and seamless integration of localized UI elements-making them ideal for EdTech platforms that anticipate future growth across different regions and languages.

- Front-end UI adjustments for different languages & text lengths.

When implementing support for multiple languages-especially those with entirely different character sets or word lengths-visual elements on your EdTech platform can often break or behave unpredictably from a styling and layout perspective.

Languages vary significantly in how they structure words and phrases. For example, German or Finnish may have extremely long compound words that don't break naturally across lines, while languages like Japanese or Chinese use compact characters that can condense significant meaning into fewer symbols.

These variations can have a major impact on how your front-end UI behaves, particularly in components like navigation bars, buttons, forms, modal windows, and card-based layouts.

To manage this, it’s common practice to create language-specific CSS files, allowing for customized styling adjustments per language. These files allow you to override or adapt spacing, font sizes, line heights, and even component widths, without affecting the global styling.

For example, in a language like German, you may need to increase padding around buttons or allow more flexible wrapping rules for headers to avoid layout breakage.

Text overflow and wrapping are particularly problematic in tightly designed interfaces where every pixel counts. A button that fits a short English label like "**Save**" might suddenly break the layout when translated to a longer phrase in another language.

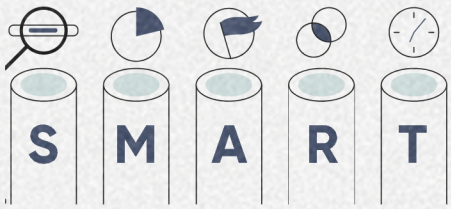
Similarly, labels in form fields, section headings, and content in tables may overflow their containers or misalign with other elements, creating a broken or confusing user experience.

By planning for these nuances from the start-using techniques like ***responsive layouts, flexible grid systems, and scalable typography***-you can ensure a smoother, more robust front-end experience across languages.

Language-specific style sheets give you the precision needed to fine-tune the UI for each audience, making your platform feel well-crafted and intentional, no matter the language displayed.

- Handling character sets, right-to-left (RTL) languages, and non-Latin scripts.

Supporting multiple languages isn't just about translation-it also involves accommodating a wide range of character sets and reading directions. Languages like Arabic, Hebrew, Persian, and Urdu require **right-to-left (RTL)** support, while others like Chinese, Japanese, Korean, and Russian involve **non-Latin scripts** with entirely different typographic characteristics.



One of the first technical considerations is your font family. Not all fonts support the full range of characters needed for these scripts. Choosing a primary font that includes wide Unicode coverage is critical to avoid rendering issues, question marks, or replacement symbols.

It's also essential to define **fallback font stacks**- so if your primary font can't render a specific character or script, the browser can fall back to a font that does. *This ensures that your platform remains visually consistent and functional*, even if the user's device doesn't have the ideal fonts installed.

For RTL languages, additional layout changes are required. It's not just about flipping the text alignment- the **entire layout often needs to be mirrored**, including navigation menus, icons, sliders, and even scroll direction in some cases.

Many modern front-end frameworks like React and Angular support RTL configuration out of the box, but implementing it effectively still requires close attention to layout logic and styling rules. CSS properties like direction, text-align, and float need to be dynamically applied based on the language in use.

Additionally, non-Latin scripts may have different typographic requirements in terms of **line height, spacing, and font sizing**, which can impact readability if not properly adjusted.
For example, Chinese characters may require more vertical space, while Arabic text might need more generous line spacing due to its cursive nature.

By planning for these complexities early on-choosing the right fonts, building layout flexibility into your front-end, and testing with real content in each language-you can ensure your EdTech platform delivers a high-quality, accessible experience for all users, regardless of the script or reading direction.

- Automating dynamic language switching & content caching.

When it comes to language localization, automation sounds appealing-but in practice, not all automation is beneficial. We **don't recommend automated machine translation** (like on-the-fly Google Translate) for EdTech platforms.

While it may seem like a quick solution, it often results in inaccurate or contextually incorrect translations, especially for educational content where precision and nuance matter.

That said, automating language switching based on user context can still enhance the user experience. A good approach is to detect the user's browser or device language settings or their geographic location, and then redirect them to the appropriate version of the platform. This method respects the user's preference while ensuring they land on the most relevant content.

When implementing multilingual routing, there are three common URL structuring methods, each with distinct SEO and technical implications.

1. **Subdomain approach** (e.g., https://english.edtechplatform.com):

While technically simple to set up, this is **not ideal for SEO**. Search engines treat each subdomain as a separate entity, which can fragment your domain authority and dilute your SEO equity.
Subdomains may also require ***more complex routing logic or even separate codebases***, increasing development and maintenance overhead.

2. **Language path (forward slash)** (e.g., https://www.edtechplatform.com/en/):
This is widely regarded as best practice. It's clear to both users and search engines which language is being served, making your content easier to index and manage. It also maintains a unified domain structure, consolidating your SEO efforts across languages.

3. **Session-based switching with unchanged URLs:**

In this case, the site dynamically changes its content based on session data, cookies, or browser language, but the URL stays the same. This approach is **highly discouraged for SEO** because search engines see a single URL with different content served to different users, which can lead to indexing confusion and poor visibility in multilingual search results.

In terms of **caching**, be aware that language-specific content needs to be handled carefully. Content delivery networks (CDNs) and caching layers must be configured to cache different language versions separately.

If not, a user might end up seeing the wrong language due to an improperly cached version. Implementing language-aware caching rules, ideally by incorporating the language into the URL path, helps mitigate this.

By combining intelligent user detection with SEO-friendly URL structures and language-aware caching, you create a multilingual experience that is both user-friendly and technically robust-laying the groundwork for sustainable international growth.

- Integration of multilingual text messages, push notifications, emails, and chat support.

Once your EdTech platform supports multiple languages on the front end, it's essential that your user communications follow suit.
Whether it's transactional text messages, emails, push notifications, or in-app chat support, **users expect a consistent experience in their preferred language**-and delivering this requires both technical planning and content preparation.

Email Communication

Multilingual email support functions similarly to templating systems on the platform itself. Each email type (e.g., welcome email, password reset, course reminder) should be created and stored in all supported languages, either within your email service provider (like Mailchimp, SendGrid, or Postmark) or in your database.

When an email is triggered, the platform checks the user's preferred language, which is typically derived from their session, profile settings, or initial onboarding selection, and sends the correct version of the email accordingly.

This ensures that even automated transactional messages are delivered in a language that feels personal and contextually appropriate.

Text Messages and Push Notifications

Text messages and push notifications also require language-aware handling. Unlike emails, these notifications are shorter and often action-driven, so clarity is key.

The user's **language preference must be stored in their profile**, and every notification should be templated per language. When a text message or push notification is triggered (for instance, a reminder to complete a lesson), the system should dynamically select the appropriate language template and **send the right message content via API** to the push notification service.

This avoids confusion and improves engagement, especially in educational contexts where instructional accuracy is essential.

Chat Support

Multilingual chat support depends heavily on the tools or chatbots you integrate. Most modern chat platforms allow you to configure the initial language of the chat interface based on the user's session or browser language.

Some advanced chatbots-especially those powered by AI-can even detect the user's language in real-time and respond accordingly

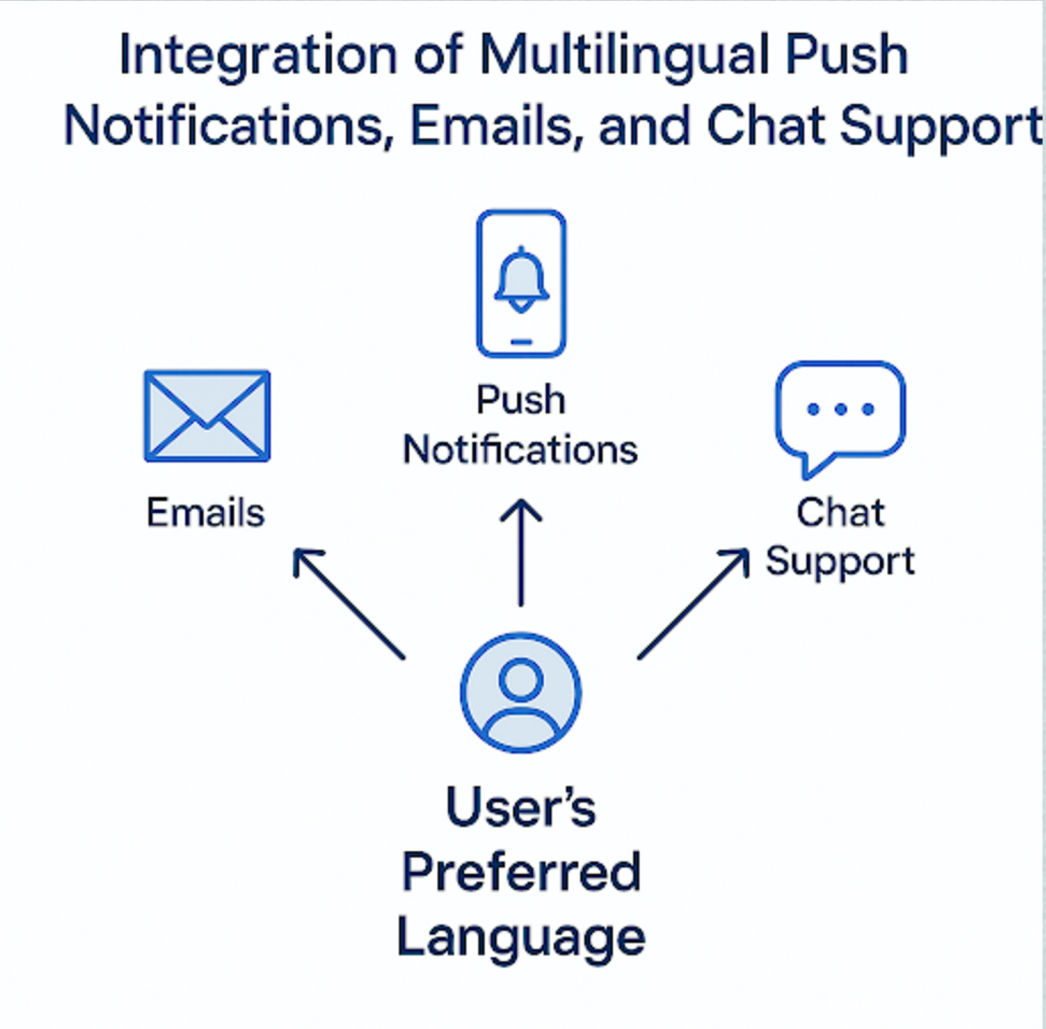
However, this still requires your team to preload and structure the content in multiple languages to maintain quality and tone.

For live chat, your support team should ***be equipped with translation tools or staffed appropriately*** to handle multilingual queries if you're operating across regions.

Mobile Apps

On mobile platforms, users typically select their language preference during onboarding or rely on the device's default language settings. Again, this preference should be stored and respected across all communication channels-emails, notifications, in-app prompts, and support-ensuring a consistent user experience.

As always, it's significantly more efficient to **build multilingual support into the mobile app architecture from the beginning**. Retrofitting language support after launch is often time-consuming and introduces the risk of fragmented messaging and inconsistent UX.



3.2 **Localization Challenges**

- Preparing content for multiple markets (& why retrofitting doesn't work)

Many EdTech companies begin with an *English-first approach*, assuming they can simply translate everything later. But in reality, translation is only one aspect of the process.

Languages differ in styles and even direction, meaning these variations can disrupt your platform's UI if localization isn't planned from the start. Plus, terminologies and cultural references may not translate directly, and even your software functionalities must be tailored for different linguistic and cultural needs.

If your product isn't internationalized to support localization from the start, retrofitting becomes expensive. Sooner or later, you will need to go back into your codebase, redesign workflows, and reengineer functionality, costing your team time, resources, and momentum.

A better idea to avoid all these challenges is to implement localization from day one, ideally within your software development sprint.

- Managing translations efficiently to avoid manual errors & version issues

Imagine updating your entire EdTech platform in all your target languages every time you roll out a new feature or lesson. Manually tracking versions and ensuring the UX is consistent can quickly turn complex.

Thanks to technological advancements, you don't need to do this manually anymore.

Powerful translation management systems (TMS) available in the market can integrate within your platform to localize every update automatically.

However, relying solely on machines for localization can lead to inaccuracies or cultural misalignment. **After years of experience in the language industry**, we've concluded that the most effective approach combines technology with human expertise, giving you the best of both worlds.

At Language Inspired, we developed a hybrid approach called **AI Translation Post Editing (AITPE)**. This workflow leverages cutting-edge AI translation tools to handle large-scale localization instantly.

But instead of stopping there, our professional native linguists fine-tune the output by correcting subtle errors, ensuring cultural relevance, and adapting the content to meet the expectations of each audience.

- Adapting educational content & assessments to cultural and linguistic contexts

We just touched on how localization goes beyond literal translation. Here's a simple example: a math problem in the U.S. might reference the number of hamburgers. In Japan, students might relate better to a statement involving sushi.

So you see, ***it is all about adapting your educational content so it makes sense culturally.*** This applies to everything from course examples and metaphors to visual elements and even humor. What's appropriate in one culture might be offensive in another.

It also comes with other features like glossaries, AI translation, and translation memory that help maintain consistency across lessons, exams, and support materials.

Assessments also need localization. A standardized test designed for English speakers may not translate well into other languages due to different phrasing, numbering systems, or even question formats.

Ensuring your content and assessments spread knowledge fairly across languages takes careful planning. A great way to do this is to collaborate with local experts who understand how students in different regions learn and process information.

- Ensuring compliance with regional education regulations

Education is one of the most regulated industries worldwide. Every country (and sometimes individual states or provinces) has its own laws governing digital education, data privacy, and accessibility requirements.

For example, accessibility laws in the USA, like WCAG, require Edtech platforms to be usable for students with disabilities. When you plan to expand globally, you must understand and adapt to these regulations to function properly.

This can become a big challenge without local industry and legal expertise that is relevant in your target region. If not done right, it can lead to costly mistakes or, worse, close the doors for your business in that market.

So, it's a good idea ***to make sure your EdTech platform meets these regulations*** from the very start or collaborate with professionals who do it for you.

Best Practices for Multilingual EdTech Platforms

04

- Plan for localization from day one (rather than adding it later).

One of the most critical decisions you can make when building a multilingual EdTech platform is whether to accommodate localization from the outset or retrofit it later. While it may seem easier to defer multilingual functionality until user demand justifies it, doing so can lead to substantial technical debt.

Planning for localization from day one ensures your platform is structurally ready-this includes everything from database design, content models, and URL routing to front-end layout and integration touchpoints.

By laying the groundwork early, you avoid costly rework, breakages in the UI, and compatibility issues with third-party tools that aren't language-flexible.
Retrofitting multilingual support after launch is not only time-consuming and expensive-it often requires untangling hardcoded text, adjusting layouts for text expansion or RTL languages, and overhauling integrations. Simply put: it's like trying to add another floor to a house without planning for it in the foundation.

- Separate UI content from code to streamline translations.

One of the most important technical decisions when building a multilingual EdTech platform is to separate all translatable content from your core codebase.

This means storing all text elements-such as navigation labels, button copy, field names, error messages, onboarding prompts, tooltips, headings, and even dynamic content like email or push notification templates-in a structured set of language files or translation variables.

By doing this, you create a centralised, scalable system that defines every string of UI text per language. Not only does this dramatically simplify the process of switching between languages, but it also makes it easier to edit or update copy later without touching the logic or functionality of your application.

When it's time to add a new language, your team won't need to search through the codebase for hard-coded strings. Instead, you simply add translations to the existing language file structure, *reducing the risk of bugs and ensuring consistency across the platform.*

This approach also allows non-developers-such as translators or content teams-to work on language-specific content without developer involvement, speeding up localization workflows and ensuring accuracy.

- Choose scalable translation workflows (manual + AI-assisted).

How well you localize your Edtech platform depends on the workflow you set for the process. While a fully manual process is time-consuming, an AI-only approach can miss literal accuracy and cultural nuances.

Rather, **practice a hybrid approach that leverages AI for speed and cost efficiency while still incorporating human expertise** for the final human touch that ensures precision and contextual adaptation.

- Test for visual accuracy in your edTech platform

When localizing an EdTech platform, it's not enough to rely on automated string replacement or assume that translations will simply 'slot in.' Every language version must be thoroughly tested-not just for visual accuracy, but for functional consistency and usability across the entire user journey.

Visual testing should include checking that all elements are correctly styled, that spacing remains consistent, and that text wraps appropriately within design constraints.

Languages with longer words or different script directions (like Arabic or Chinese) often push the limits of pre-defined layouts. Watch out for broken layouts, overlapping

elements, or truncated content-especially in places like buttons, navigation menus, table columns, and mobile views.

Functional testing goes beyond what's on the screen. Every interactive element needs to be verified across all supported languages-this includes *form validation messages, user dashboards or portals, account creation flows, in-app notifications, emails, and push notifications*.

Each of these should reflect the correct language, pull the appropriate content templates, and behave exactly as expected.

Keep in mind: some issues only surface in live or real-user environments (e.g. incorrectly formatted dates, unsupported characters in input fields, or unexpected third-party tool behaviour), so end-to-end testing in each language context is essential.

Actionable Steps to Implement Localization & Multilingual Development

05

1. Assess your current platform's localization readiness

Before you begin implementing multiple languages, it's crucial to conduct a thorough **code and architecture audit** to assess whether your platform is built to support multilingual functionality-or whether it needs foundational changes to accommodate localization properly.

This assessment focuses on the **technical readiness** of your platform, not just whether translation files exist, but whether the underlying infrastructure can scale to support multiple languages efficiently.

What the audit covers:

- Code structure: Is all front-end content hardcoded into templates, or has the platform been set up with reusable translation variables and a defined structure for localization?
- Text separation: Are UI labels, messages, and content properly decoupled from code logic?
- Routing and URLs: Does the site structure allow for language-based routing (e.g. subfolders like /en/, /fr/)? Are routes dynamically configurable based on locale?
- Database readiness: Are there provisions for storing multilingual content (e.g. product titles, descriptions, user-generated content) in a scalable way?

- Third-party services: Are APIs, plugins, and integrations compatible with multilingual support, or do they contain hardcoded text that will need overrides?

- Language detection mechanisms: Are there any mechanisms in place to detect user language preferences and switch content accordingly?

The result of this audit is a detailed technical report that outlines:

- The current localization status of your platform (fully ready, partially ready, or not ready)

- Identified gaps and limitations in your codebase, architecture, or integration points

- Risks of proceeding without restructuring

- Prioritised next steps for enabling or improving multilingual support

- Recommendations for tools, frameworks, or libraries to use in future implementations

- This assessment is a vital first step-it gives you the technical clarity to plan your localization rollout with confidence, **ensures you avoid costly retrofits later**, and provides a blueprint for integrating additional languages smoothly as your platform grows.

Beyond technical platform considerations

- Beyond technical aspects, review your original content to ensure it's ready for localization. Start by identifying specific references that may need cultural adaptation. It's also a good idea to standardize terminology using glossaries as it helps maintain consistency across languages. Finally, make sure you take into account different aspects of your platform ***beyond text, including the UI, payment methods, date and currency formats.***

2. Define your language expansion strategy

- Simply launching a multilingual Edtech platform won't yield the desired results if it doesn't align with your business goals and user needs. People aren't going to suddenly trust it and turn into customers. If you put yourself in their shoes, you'd feel exactly the same way.

- Rather, if you want to scale your business into new markets, first conduct extensive research. Again, this means thoroughly understanding your audience, understanding your competitors, and also building a relevant brand image based on your research.

Ask yourself these questions:

—> What are the markets that are bringing you the highest traffic?

—-> What are the most commonly spoken languages in these regions?

—-> Who are the existing competitors in those regions and what are they offering?

Once you have a fair idea about your markets, create a phased rollout plan, ideally starting with the largest user bases and markets that present the highest revenue potential. Remember to establish clear KPIs per language that matter the most.

3. Implement the right development structure

Once your platform's localization readiness has been assessed, the next step is *to implement a development structure that can support multiple languages efficiently* - both now and in the future.

a. **Database Setup for Multilingual Content**

From the ground up, your database should be structured to store content in multiple languages without redundancy or confusion. This means:

- **Using language**-specific fields or tables to store translatable content like page titles, descriptions, product names, and user-facing messages.

- **Avoiding duplication** - don't create an entirely new record for each language, but rather extend existing structures to support language variations.
- **Supporting fallback logic** - if a piece of content isn't available in the selected language, the system should gracefully default to a fallback language (typically English).

A well-planned schema allows for easy data management, clean retrieval, and seamless integration with localization libraries and APIs.

b. **UI Flexibility for Multilingual Layouts**

Your front-end should be built with flexibility in mind. Different languages bring different challenges (which we've touched on already), such as longer words, different character sets, and entirely different layout directions (like RTL).

To handle this well:

- Use dynamic layouts that can accommodate content of varying lengths without breaking designs.

- Define language-specific CSS overrides for spacing, font size, alignment, or special UI treatments. For complex platforms, it's often worth having per-language CSS files.

- Ensure components are responsive and adaptive across screen sizes and devices - this becomes especially important when translated strings are significantly longer.

- Avoid hardcoded widths or truncations - instead, allow containers and layouts to grow with content.

By combining robust database design with front-end flexibility, your EdTech platform will be well-positioned to handle localization gracefully, without compromising performance or design integrity. **This foundation reduces the complexity of adding new languages down the line**, and ensures a consistent user experience regardless of locale.

4. Adopt an efficient localization process (Tools, workflows, automation).

We mentioned this earlier and we will emphasize it again — it is necessary to have an effective localization workflow that scales with your product.

To give you a clear picture, it starts with developing your platform in a way that's ready to localize and preparing content in a way that's easy to translate. From there, translations go through an automated process where AI does the initial job and then **linguists**, **developers**, and **QA teams** work in sync to deploy the final, refined output.

Yes, the key is to make localization an integrated, ongoing process in sync with your development sprints rather than a last-minute add-on.

5. Conduct rigorous testing

Test, test, and test some more, before deploying the localized versions. When you do, here are some elements to consider:

- **Test for Language Accuracy.**

Does the translation make sense in the learning environment? Are idioms, examples, and references adapted to the local audience? As an Edtech business, *your translated content will be the biggest pillar supporting your global expansion*. So, involve native experts who can assess not just correctness but readability and engagement of your content before you roll it out for real learners.

- **Test UI and Layout for Multilingual Responsiveness.**

German and Russian texts can be 30-40% longer than English, and when that happens, your UI must adapt. It goes for multiple other aspects, starting from text, graphics, buttons, menus, and even error messages. It's important **to test the UI across different devices**, such as desktops, laptops, tablets, and mobile phones.

- **Functional Testing Across All User Journeys.**

Localization affects not just content but also platform functionality. So, every localized version must be tested for user inputs and validations, third-party integrations, notification systems, chatbots, and everything else that triggers an action within your platform.

- **Simulate Real User Experience (Linguistic & UX Testing).**

Lastly, conduct real-world usability tests, ideally by native-speaking testers or beta users. These testers interact with the platform as actual learners, providing feedback on content clarity, navigation ease, engagement factors, etc.

6. Plan for long-term scalability

A well-architected multilingual platform is one that can scale - not just in terms of users or features, but also in terms of languages, markets, and complexity. Localization shouldn't be something that breaks or slows down your platform as you grow. Planning for long-term scalability ensures that new languages, tools, and content types can be introduced without costly rework or painful retrofitting.

Design for Any Language - Not Just the Ones You Know Now

Even if you're only launching in one or two languages to begin with, your **system should be structured to accommodate any language** - including those with different scripts, directionality, or unique formatting rules. When the web app is properly planned and modular from the outset, introducing new languages becomes a simple process of adding translation files and content, not overhauling your codebase.

Build Language Support into New Features by Default

Every new feature or section of the platform should be developed with multilingual support in mind from the start. This includes:

- UI components
- Forms and error messages
- User flows (e.g. onboarding, checkout, account setup)
- Notifications and system messages

By making localization part of your definition of done, you avoid having to "patch in" multilingual support after a feature is live.

Evaluate Third-Party Tools for Multilingual Compatibility

Many web and mobile platforms rely on third-party APIs, SDKs, and widgets for functionality like payments, chat, video conferencing, or user analytics. When integrating these, it's crucial to:

- Assess what language support they offer
- Understand whether they can inherit your platform's language settings
- Identify where manual overrides or language-specific customisations may be needed (e.g. localising labels or UI text rendered by the integration)

This proactive evaluation avoids unpleasant surprises where parts of your product remain untranslated or inconsistent.

Stay Compatible with Browser and OS Updates

While most language settings are handled cleanly by modern browsers and mobile OSs, major updates can sometimes cause layout shifts or encoding issues.

As a best practice, schedule testing of multilingual flows after significant updates to iOS, Android, Chrome, Firefox, etc., particularly if your platform has a mobile component or progressive web app (PWA).

Scalability Comes from Foundation, Not Fixes

If best practices are followed early - from database design and translation management to component styling and content strategy - then scaling your platform for new languages, regions, and users should be a smooth, repeatable process.

What you want to avoid is building something that works today but creates mounting technical debt with every additional language or market you add.

Takeaway: Localization isn't a one-time task. It requires continuous management as your platform evolves. As new features, courses, and curriculum updates roll out, your localization workflow must keep pace without disrupting the user experience.

To do this in the most efficient and accurate way possible, *we suggest you leverage cutting-edge technology like AI combined with human expertise*.

Case Studies & Success Stories

06

- Examples from Elemental (EdTech platforms with scalable architectures).

https://www.elemental.agency/our-work/disco-interactive-online-learning-platform-web-development

https://www.elemental.agency/our-work/sure-trading-educational-trading-platform-development

https://www.elemental.agency/our-work/ukesa-bespoke-php-web-app-wireframe-design-development-project

- Language Inspired experience in educational localization.

Problem
XtraMile, a Norway-based learning automation platform, faced recurring issues with poor translation quality. As a result, they misspent resources and the product rollouts ultimately got delayed.

To align with their tech-forward vision, they sought a smarter, AI-driven localization workflow that matched their product's innovative nature.

Solution
Language Inspired introduced an AI-powered translation post-editing (AITPE) workflow tailored for their English (EN) > Danish (DA) needs.
The new workflow leveraged AI capabilities for faster, accurate translations.

It also incorporated Translation Memory for precision, the company's Style Guide for consistency, and expert human post-editing to ensure the highest quality output.

Approach

Our native linguists *conducted deep research* into XtraMile's brand, product, and industry to create a term base and style guide. Using Lokalise as the client's chosen

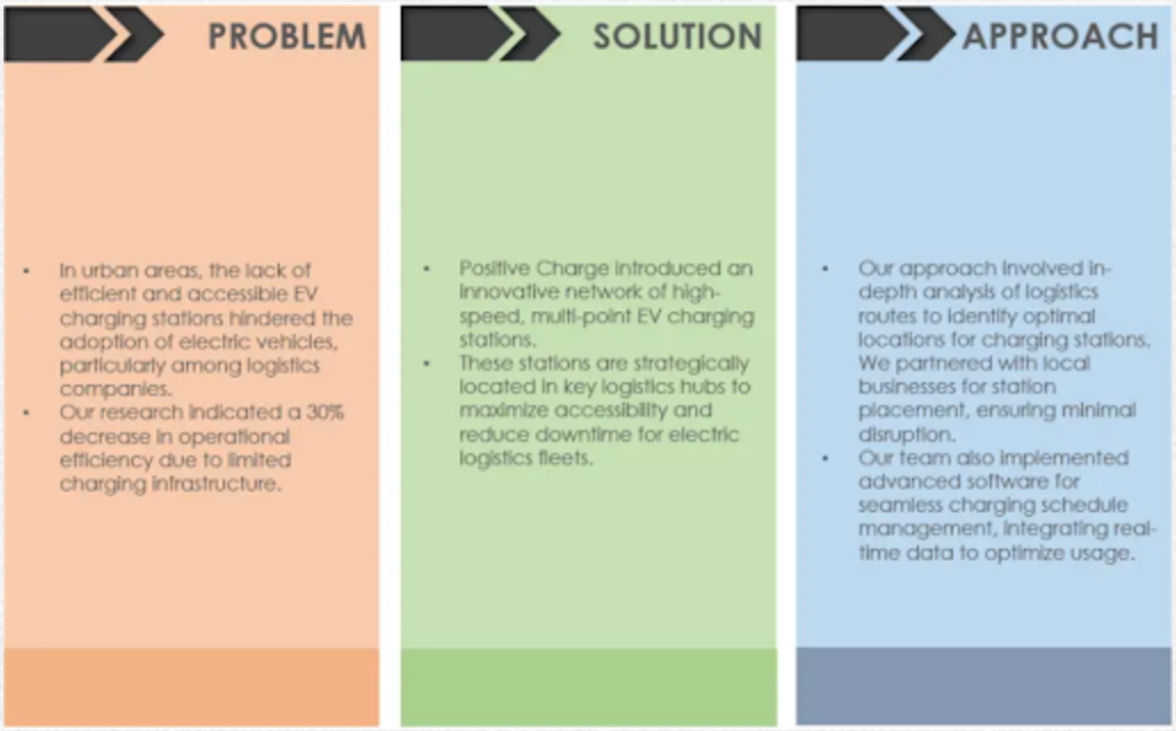
TMS, we then built a custom workflow integrating brand-specific linguistic assets, glossary, and stylistic rules within the platform.

Throughout the process, we closely collaborated with XtraMile's CEO and Head of Software Development to manage scopes, deadlines, and priorities. Every time new product strings were added, ***Language Inspired executed the localization work in sync with their development cycle***.

Translations were validated in their live product environment to ensure they were accurate, functional, and fully on-brand before release.

Impact

Within weeks, XtraMile delivered the Danish version of their e-learning product for a particular end-client, who provided exceptional feedback and continues to use it regularly. Our localization solution restored XtraMile's confidence in outsourced localization and seamlessly aligned with their agile development cycle.



"I'm impressed with this first round. Language Inspired team was very thorough with going through the app."

– XtraMile CEO

- Hypothetical success scenario showcasing a seamless collaboration.

A UK-based EdTech company plans to launch its online learning platform in three markets: the **UK** (home market), the **USA**, and **LATAM**. Their goal is to expand fast without compromising product quality, user experience, or cultural relevance.

The company has a solid MVP built for UK users, but lacks the technical and linguistic infrastructure to scale internationally. They need:

- A platform architecture ready for international use
- Region-specific UI/UX adjustments
- Multilingual content tailored to learners in the UK (British English), USA (American English), and LATAM (Spanish and Portuguese).
- Rapid product development and deployment aligned with their growth roadmap

Elemental, with its expertise in software development, ***re-engineers the Edtech platform*** to support internationalization (i18n) at its core.

The architectural designs are made modular, the backend code is modified, and the front-end components are made flexible to accommodate any language (for now and the future). Integrations are also taken care of to support localization.

In parallel, Elemental partners with Language Inspired to handle the end-to-end localization process of the platform. The LI team develops a terminology base and brand voice guidelines per region.

They also create a custom workflow that integrates within the SDLC to optimize the product's time-to-market. The team uses advanced AI technology and human linguists to preserve brand consistency and linguistic accuracy while delivering on-time localization.

As a result of their collaboration, **the platform can launch in all three markets within the next three months - fully functional and localized**. The company also sees 10x user engagement, positive feedback, and a shorter time-to-market for new updates.

Want similar results for your Edtech platform or want to learn more? Talk with our expert and we will be happy to help.

Conclusion & Next Steps

07

- Why localization drives global EdTech success

Reaching a global audience in EdTech isn't just about translating content-it's about creating inclusive, accessible experiences that resonate with diverse learners, across different tech platforms and devices.

Multilingual capabilities allow platforms to scale into new regions, improve learner outcomes through localization, and foster deeper engagement by meeting users in their native languages. In a competitive and increasingly borderless market, multilingual functionality is no longer a nice-to-have-it's a key driver of global growth and success.

- **How Elemental & Language Inspired can help: Our unique partnership advantage**

Elemental and **Language Inspired** combine deep technical expertise with proven localization experience to offer EdTech companies a powerful, end-to-end solution.

From designing and developing scalable, multilingual web and mobile platforms, to implementing seamless localization workflows and high-quality translations-our partnership bridges the gap between technology and cultural nuance.

- Together, we help EdTech companies expand their reach, elevate their product experience, and grow confidently into new markets.

- Let's discuss your localization needs!
- Call-to-action:

- Contact us for a free consultation on your multilingual EdTech needs.

Want to know more about how Language Inspired can help your business? Book a call with our team and see why EdTech companies trust our service.

GET IN TOUCH

LANGUAGE INSPIRED

Speak local. Connect global.