



MENU

INTRODUCTION

EXISTING METHOD

PROPOSED METHOD

METHODOLOGY

IMPLEMENTATION

CONCLUSION

LANGUAGE TRANSLATOR

LANGUAGE TRANSLATOR APP

Language translator using PHP





ABSTRACT

Language translator is a php based web app which is used to translate different language.

Unlike other application it has certain advantages:

- It can be **easily access** by everyone having internet access.
- It **detects the language automatically** entered by the user.
- Can be **converted into any of the desired language**.
- **Beautiful** and **simple UI**.
- **Handy** to use.

Link to the Repository : <https://github.com/Abhi6722/language-translator>

Link to the Application : <https://language-translator-abhi6722.herokuapp.com>





TABLE OF CONTENT

01

Introduction

02

Existing Method

03

Proposed Method

04

Methodology

05

Implementation

06

Conclusion





MENU

INTRODUCTION

EXISTING METHOD

PROPOSED METHOD

METHODOLOGY

IMPLEMENTATION

CONCLUSION

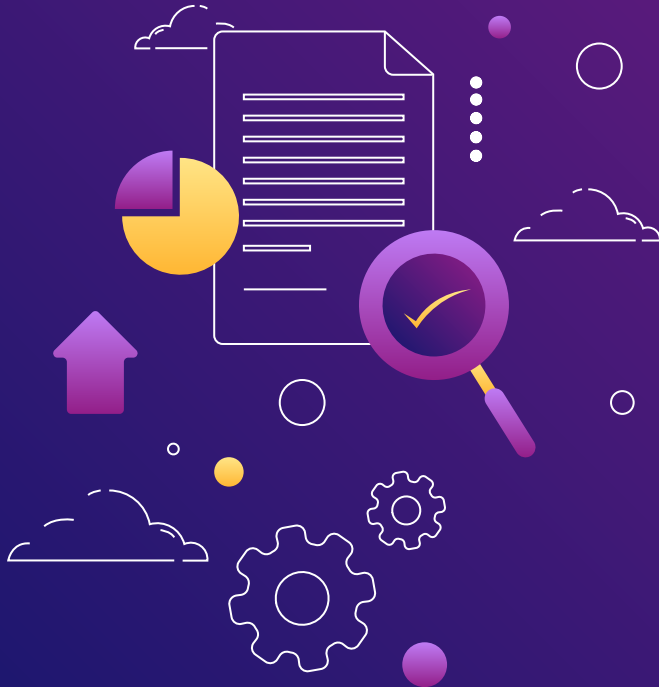
LANGUAGE TRANSLATOR

01

INTRODUCTION TO PROJECT

Here is the introduction of the project





INTRODUCTION

The task was to create a language translator app or web application that can translate from English to any language. There were various approach to try out this project but since I am familiar with php, html, css, js so I used these languages to create a translator app. See the website live [here](#).





MENU

INTRODUCTION

EXISTING METHOD

PROPOSED METHOD

METHODOLOGY

IMPLEMENTATION

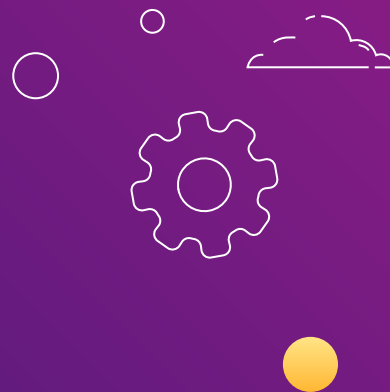
CONCLUSION

LANGUAGE TRANSLATOR



EXISTING METHOD

Let's discuss the existing method of app





EXISTING METHOD VS SOLUTION



EXISTING METHOD

There were various existing methods available which can be used for translation

VS

SOLUTION

In our solution main aim included auto detection of language along with beautiful and easy UI.





UNDERSTANDING DATA



INPUT

Once user enter the text it is automatically detected in the application or user can select from the menu.



OUTPUT

For the output user has to select the desired language in which output is required.





MENU

INTRODUCTION

EXISTING METHOD

PROPOSED METHOD

METHODOLOGY

IMPLEMENTATION

CONCLUSION

LANGUAGE TRANSLATOR

03

PROPOSED METHOD WITH ARCHITECTURE



Let's discuss the proposed method





PROPOSED METHOD WITH ARCHITECTURE

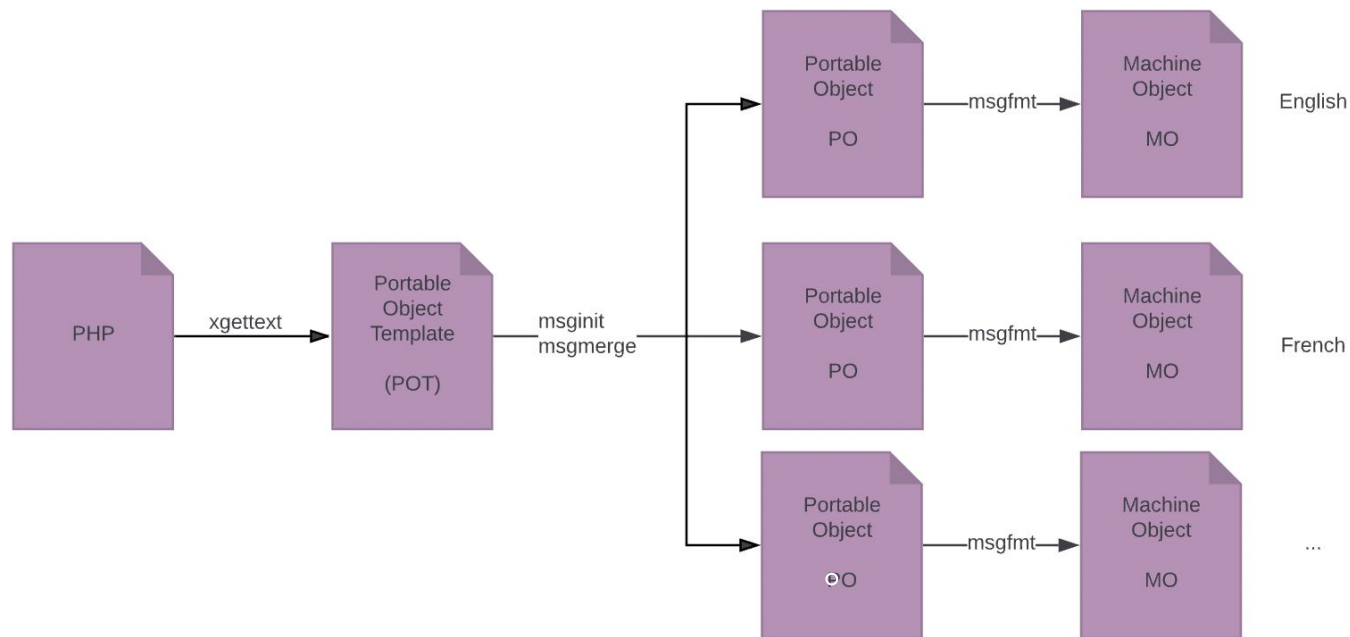
The following image shows the translation workflow for app. The basic steps are:

1. Extract messages to translate with xgettext, stored as Portable Object Template file (.pot)
2. Use msginit and msgmerge to create a Portable Object file (.po) for each language
3. Convert the PO files into optimized Machine Object Files (.mo) using msgfmt





gettext workflow for PHP





DEPENDENCIES



PHP

PHP version 8.1.0 is used
in application

ext-mbstring

Ext-mbstring bundled with
php

composer

Composer 2.1.14 is used
in application

apache

Apache version of 2.4.51
is used

nginx

nginx version 1.20.2 is
used in project





METHODOLOGY

Create translation files

1. Extract strings with xgettext

With the xgettext tool you can extract all translatable strings from the PHP source code, store them in a PO template file with file extension .pot. Don't edit the POT file - xgettext overwrites it each time you start it.

```
xgettext --add-comments *.php myapp.pot
```

With the --add-comments option comment blocks preceding the `_(...)` expression are copied from the source code to the .pot file. With such comments, you can give a hint to the translator how he should translate the text.





2. Create initial PO files with msginit

The POT file is the template file used to create a new PO file for each target language. Place them in directories as mentioned above:

```
msginit --locale de --input myapp.pot --output locale/de/LC_MESSAGES/myapp.po  
msginit --locale fr --input myapp.pot --output locale/fr/LC_MESSAGES/myapp.po
```

This step is only needed once when you set up your project. As `msginit` overwrites an existing PO file, you shouldn't call it if you have already PO files containing translations. In this case use `msgmerge` as described later.

3. Translate the PO files

Now you can start translating the PO files. You can use a text editor or a special translation editor for that. If your translations contain non-ASCII characters, change the charset in the header from ASCII to UTF-8:

```
"Content-Type: text/plain; charset=UTF-8"
```





4. Convert PO to MO files using `msgfmt`

You next have to convert the PO files into "Message Object" files (with the extension `.mo`) which can be loaded on the server:

```
cd locale/de/LC_MESSAGES/  
msgfmt myapp.po --output-file=myapp.mo
```

5. Enjoy

Start your PHP script and enjoy its translated output. Keep in mind that `gettext` caches the translation data. If you add/remove/update MO files you might have to restart/reload your web server.

6. Managing changes with `msgmerge`

If you're updating your PHP code, you might change translatable texts, too. This requires an update of the PO/MO files. There's, of course, a `gettext` tool to simplify this task, it's called `msgmerge`. First, you have to extract the translatable strings from the new PHP sources using `xgettext`. This overwrites your old POT file with a new one. Then you can merge the changes into the language-specific PO files:





```
xgettext --add-comments *.php myapp.pot msgmerge --update locale/de/LC_MESSAGES/myapp.po  
myapp.pot ... translate new / changed texts in myapp.po ...
```

msgfmt locale/de/LC_MESSAGES/myapp.po --output-file=locale/de/LC_MESSAGES/myapp.mo
msgmerge doesn't delete any translation: If a string is no longer found in the source code, it is marked as deleted in the PO file:

```
#~ msgid "Draft"
```

```
#~ msgstr "Entwurf"
```

New strings are added to the PO file, for unchanged strings the source code line number, and the extracted comment are updated, if necessary.

`msgmerge` also tries to detect changed strings using a fuzzy string comparison. If only a small part of the string has changed the new string is saved together with the old translation in the PO file, and the entry is marked as "fuzzy".





Example: If you change "Translation of PHP pages with gettext" to "Translate your PHP pages with gettext" you will get this PO file entry. The translation is outdated and should be updated in the next step:

#: index.php:17

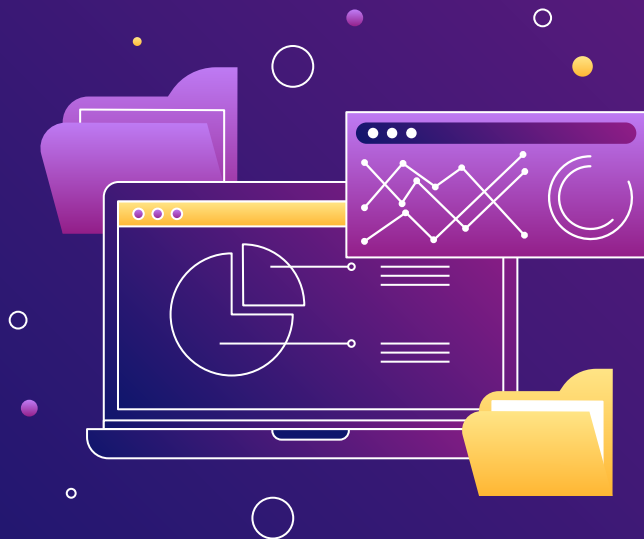
#, fuzzy

msgid "Translate your PHP pages with gettext"

msgstr "Übersetzen von PHP-Seiten mit gettext"

If `msgmerge` cannot detect a similarity between old and new string, the old one is marked as deleted, and the new one is added as new entry, without translation.





IMPLEMENTATION

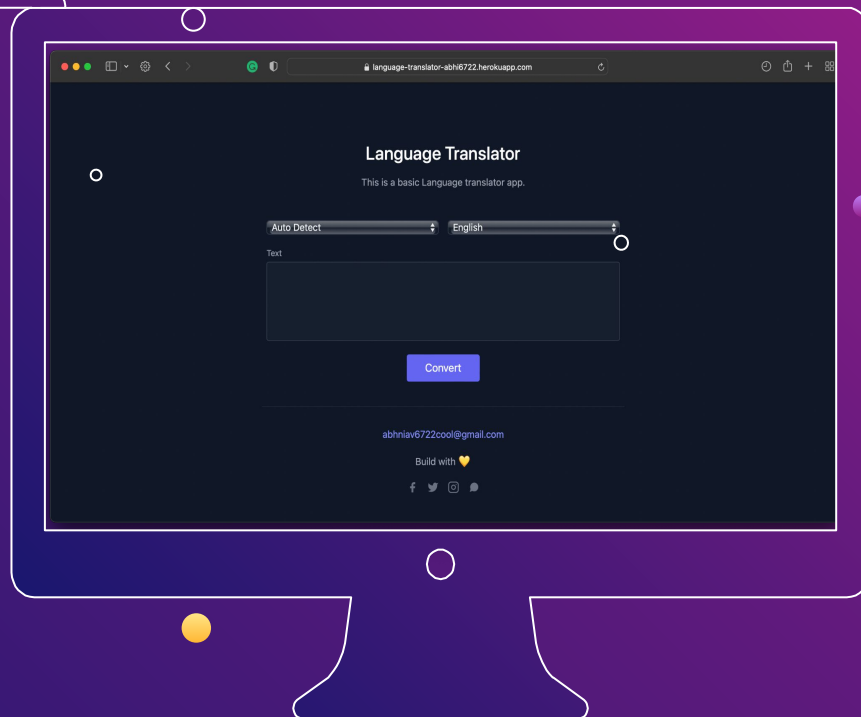
Since the main objective from the start was to make it easy to access from anywhere so deploying the project was an important phase. First the project was successfully uploaded on Github and then it was deployed to Heroku.

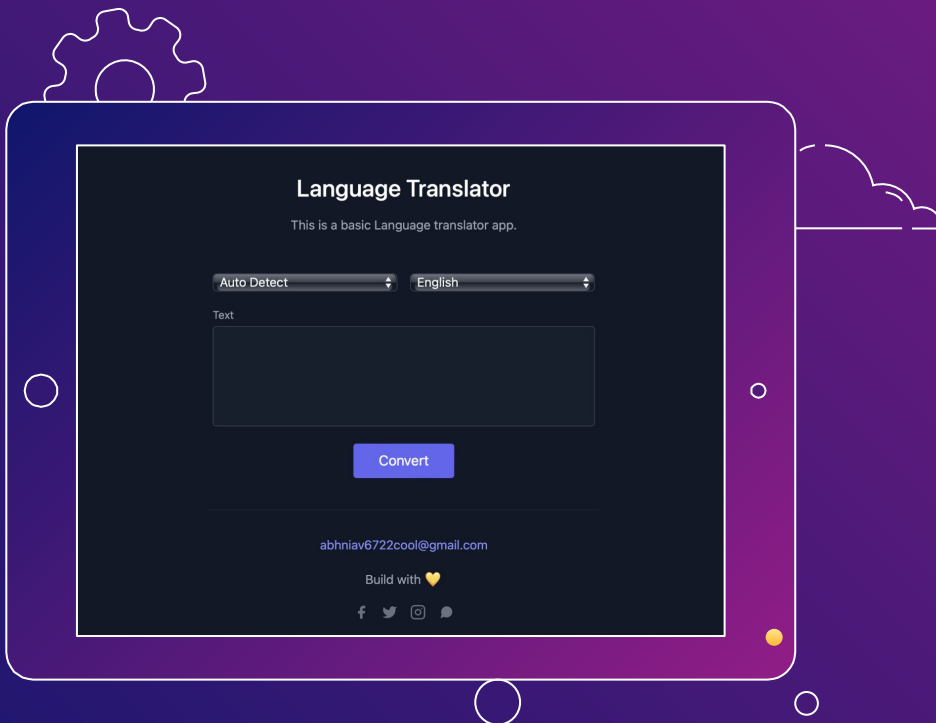


[MENU](#)[INTRODUCTION](#)[EXISTING METHOD](#)[PROPOSED METHOD](#)[METHODOLOGY](#)[IMPLEMENTATION](#)[CONCLUSION](#)[LANGUAGE TRANSLATOR](#)

DESKTOP VIEW

This is a Desktop version of the application and it is completely responsive.



[MENU](#)[INTRODUCTION](#)[EXISTING METHOD](#)[PROPOSED METHOD](#)[METHODOLOGY](#)[IMPLEMENTATION](#)[CONCLUSION](#)[LANGUAGE TRANSLATOR](#)

TABLET VIEW

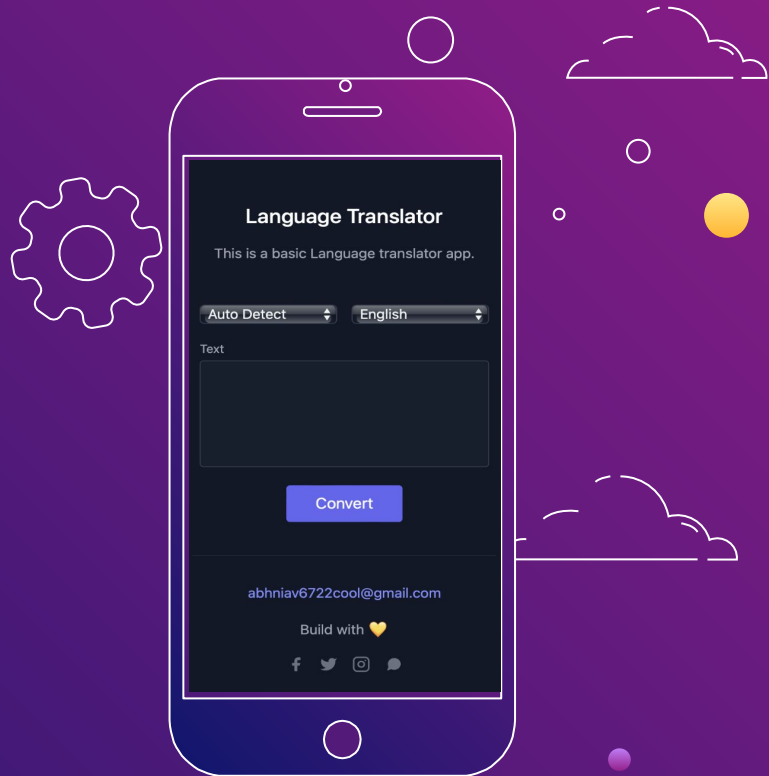
This is a tablet version of the application and it is completely responsive for tablet device.



[MENU](#)[INTRODUCTION](#)[EXISTING METHOD](#)[PROPOSED METHOD](#)[METHODOLOGY](#)[IMPLEMENTATION](#)[CONCLUSION](#)[LANGUAGE TRANSLATOR](#)

MOBILE VIEW

This is a Mobile version of the application and it is completely responsive to all mobile devices.





CONCLUSION

The Language Translator web app is successfully created and deployed to Heroku [here](#). The main objectives were also fulfilled as the UI is easy to use and also Beautiful along with various language provided.

Link to the Repository : <https://github.com/Abhi6722/language-translator>

Link to the Application : <https://language-translator-abhi6722.herokuapp.com>



[MENU](#)[INTRODUCTION](#)[EXISTING METHOD](#)[PROPOSED METHOD](#)[METHODOLOGY](#)[IMPLEMENTATION](#)[CONCLUSION](#)[LANGUAGE TRANSLATOR](#)

THANKS!

DO YOU HAVE ANY QUESTIONS?

abhinav6722cool@gmail.com

+918292459672

www.abhi6722.in

