

WROCLAW UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

FACULTY OF ELECTRONICS  
EMBEDDED ROBOTICS

INTERMEDIATE PROJECT

---

Web application for learning vocabulary  
implemented in Django

---

*Author:*  
Daria NOWICKA

*Supervisor:*  
Witold PALUSZYŃSKI, Ph.D.

**Abstract**

The goal of the project was to implement web application for learning foreign language vocabulary. The idea was inspired by Duolingo platform ([www.duolingo.com](http://www.duolingo.com)). The main assumption was that created application gives similar possibilities as Duolingo, but is not an exact copy of this popular website. All the established goals were achieved and even some additional functionality was accomplished. The project gives a lot of possibilities of further development.



*This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.*

January 27, 2017

# 1 Introduction

## 1.1 Goals

The goal of the project was to implement web application for learning foreign language vocabulary. The idea was inspired by Duolingo platform ([www.duolingo.com](http://www.duolingo.com)). Duolingo enables learning a big number of languages in very accessible way, i.e. testing modes are quite original in comparison with other platforms dedicated for learning languages. The biggest advantage of my application over Duolingo is that as a developer, I can add to the database any words that I need to learn at the moment. This makes the application especially useful when preparing for tests during language courses.

Assumptions and planned functionality of the application were the following.

1. The application will enable learning english vocabulary. It is dedicated for users who know polish language and have basic knowledge of english, enough to navigate through the website.
2. Every user will have his own account. There, history of his test results will be remembered.
3. Vocabulary will be divided into categories. Additionally, category „all”, containing all the words from database, will be available.
4. It will be possible to repeat the test containing only words that were wrong previously.

## 2 Implementation

### 2.1 Tools

The main tool used in the development process was Django framework (version 1.10.4 with Python 3.5.2). Additionally, some web technologies were used:

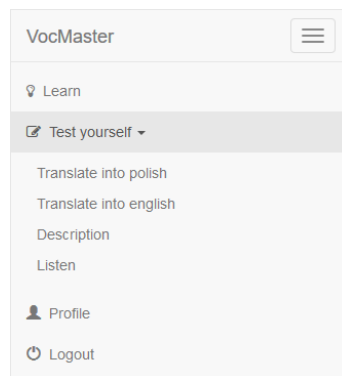
- HTML
- CSS
- Bootstrap 3.3.6
- JavaScript
- jQuery 2.2.0 (including AJAX)

### 2.2 Accomplished goals

#### 2.2.1 Appearance

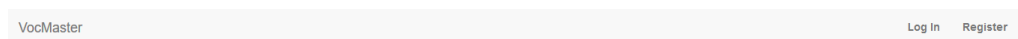
The appearance was considered as a tangential part of the project. The assumption was to make it eye-friendly, without any special visual effects. Below the goals achieved in this field are described.

1. Thanks to Bootstrap library the website is responsive (i.e. adjusts its appearance to the device where it is displayed, e.g. notebook, tablet or smartphone) (Figure 1).
2. There are separate views for different kind of pages:
  - main page,
  - page for learning,
  - pages for testing in different testing modes,
  - profile page with history of results achieved in tests,
  - logging and registering page.
3. The currently active page is signaled by the changed colour on the navigation bar.

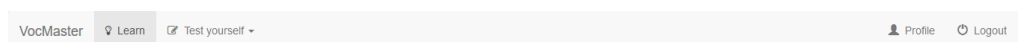


**Figure 1:** Navigation bar seen as on a mobile phone

Very important thing is, that different views are available for users logged in and different for not logged guests (Figure 2). Guests can only see the main page with short communicate and both logging and registering page (Figure 3), while logged users can learn, make test and display their results history.



(a) navigation bar for a guest



(b) navigation bar for a logged user

**Figure 2:** Navigation bar depending on a visitor type

(a) login form

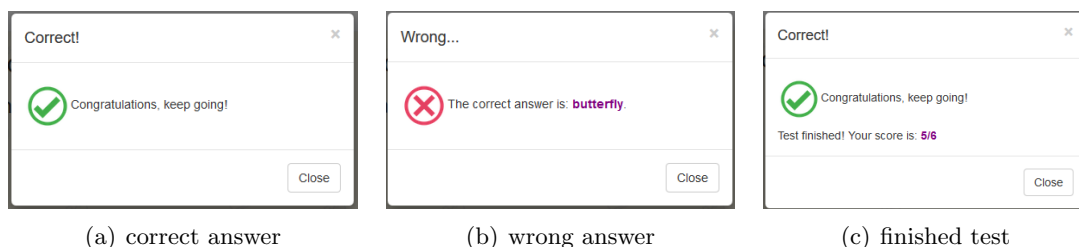
(b) register form

**Figure 3:** Login and register form

### 2.2.2 Functionality

The described application has many functionalities. Let them be introduced.

1. Possibility of registering, signing in and logging out users.
2. Basic error handling while logging in and registering (passing wrong password while logging in or username that is already taken while registering).
3. Vocabulary is divided into categories. Views with available words in each category are generated automatically.
4. Several testing modes are available:
  - translating word from english into polish,
  - translating word from polish into english,
  - writing word in english after listening to it,
  - writing word in english after reading its description.
5. Additional functionality during testing:
  - words are displayed in random order,
  - in the testing mode with the word description, there is a hint available (when hovering the mouse over a light bulb, the image of the word appears),
  - button „check” is not clickable when input field is empty,
  - after each answer a popup window appears with a message whether the answer was correct or not (if it is wrong, the correct answer is displayed) (Figure 4),
  - at the end of the test, the obtained result is shown and, behind the scene, the result is being added to the database,
  - directly after the test, it is available to repeat the test with only those words, that were wrong previously,
  - test can be repeated any number of times.



**Figure 4:** Popup windows with the verification of the answer

## 3 Summary

### 3.1 Conclusions

During the development, the application was tested by four different people. They varied in age, education and level of knowledge of english. Their remarks were very helpful while removing bugs and making the application more intuitive and easy-to-use. Usage of Django framework simplified implementation of many functionalities. The most significant feature was that Django provides a built-in database mechanism. Thanks to that fact, a lot of effort related to configuring the database from scratch was saved.

The main technology used during developing testing mechanism was JavaScript. It appeared that JavaScript provided the simplest solutions for actions like displaying words in random order of checking user inputs.

All the assumed goals were accomplished. Even some additional functionality has been achieved.

### 3.2 Possibilities of further development

In fact, possibilities of further development of described application are unlimited. The exemplary ideas of development are

- adding new words and categories,
- adding new testing modes (e.g. connect the words, fill the gaps in the sentence, arrange a word from mixed letters etc.),
- adding statistic analysis, e.g. filtering results history by different queries, generating plots and charts of progress etc.,
- expanding application, so that many languages can be learned.

The author wants to emphasise that this application is not intended to be an exact copy of Duolingo. Duolingo platform was just an inspiration, even if a strong one. According to that fact, any comparisons to Duolingo is not desired.

## 4 Resources

- Django documentation  
<https://docs.djangoproject.com/en/1.10/>
- Django tutorial on YouTube  
<https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBImzzFcLgDhKTTfNLfX1IK>
- Code School  
<https://www.codeschool.com/>
- w3schools  
<http://www.w3schools.com/>
- English words to download in mp3  
<https://ssl.gstatic.com/dictionary/static/sounds/de/0/supercalifragilisticexpialidocious.mp3>  
(type chosen english word instead the bold one to listen to it)

The source code of described application is available at my GitHub account:

[https://github.com/daria-n/Learn\\_vocabulary\\_web\\_app](https://github.com/daria-n/Learn_vocabulary_web_app)