



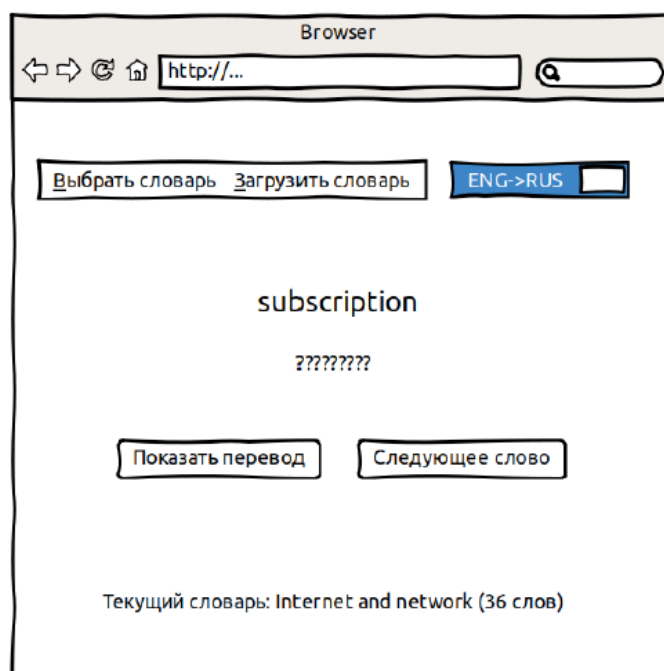
## Международная цифровая олимпиада «Волга-IT`22»

### Дисциплина «Веб-разработка (PHP)»

#### Задание отборочного этапа

Участнику отборочного этапа олимпиады необходимо разработать Web-приложение, помогающее учить английские слова.

Приложение должно выглядеть примерно так:



#### Идея работы приложения:

1. На занятиях по английскому языку пользователь получает несколько слов или выражений, которые ему необходимо запомнить. Из них он составляет таблицу в Excel или другой программе, в которой содержатся 2 столбца: в первом слова на английском языке, во втором - перевод на русский. Первой строкой должен быть заголовок с трехбуквенным кодом языка.

Пример:

ENG	RUS
honest	честный
exam	экзамен
subscription	подписка
to speak	говорить

2. Пользователь экспортирует файл в формате CSV и загружает в приложение. При экспорте файла пользователь будет указывать имя словаря.
3. Далее пользователь в меню “Выбрать словарь“ выбирает загруженный словарь. На экране появляется первое слово из словаря, а перевод должен быть скрыт.
4. Пользователь вспоминает (или не вспоминает) перевод показанного слова. Чтобы проверить, нажимает кнопку «Показать перевод», после чего перевод отображается на экране.
5. Для перехода к следующему слову пользователь нажимает кнопку «Следующее слово». На экране отображается слово, предлагаемое для перевода, а сам перевод скрывается.
6. Таким образом пользователь проходит по всем словам, проверяя свои знания. После проверки последнего слова и нажатия кнопки «Следующее слово» пользователю отображается первое слово. Таким образом все слова следуют друг за другом по кругу.
7. В любой момент пользователь может сменить направление перевода слов. Т.е. изначально я выбираю направление перевода ENG -> RUS. Приложение отображает английское слово и предлагает вспомнить перевод на русский. Если пользователь меняет направление перевода RUS -> ENG, то приложение отображает русское слово и предлагает вспомнить перевод на английский.

## Требования:

1. Приложение должно быть создано по принципу single page application, т.е. интерфейс приложения должен отрисовываться в браузере без перезагрузок страницы. Допускается перезагрузка страницы только на странице загрузки файла словаря.
2. Backend приложения должен быть написан на языке PHP версии не ниже 7.3.
3. Backend приложения должен быть написан на любом из фреймворков. Написания приложения на «чистом» PHP не приветствуются (но допускаются).
4. Frontend приложения может быть написан с использованием популярных JavaScript фреймворков, таких как: Vue, Angular JS, React, JQuery и т.д, а так же на чистом JavaScript. Допускается использование TypeScript и других диалектов JS.
5. Для реализации интерфейса можно использовать «Bootstrap» <https://getbootstrap.com/>.
6. Результат выполнения задания будет приниматься в виде ссылки на репозиторий в Github, Bitbucket или другой системы.
  - a. Репозиторий должен содержать Readme.md файл, в котором должна быть указана инструкция по сборке и запуску приложения. Инструкция должна содержать последовательность команд, которую нужно выполнить для запуска приложения.
  - b. Переменные для настройки приложения (например, доступы к базе данных) должны быть в '.env'-файле. Самого '.env'- файла не должно быть в репозитории, но должен быть '.env.example' файл, в котором все необходимые переменные будут указаны. В инструкции по установке должны содержаться инструкции по настройке этих переменных.
  - c. В репозитории не должно быть кода библиотек из composer, npm и т.д., если они используются. Только 'composer.json', 'package.json'

файлы. В инструкции по сборке должны быть указаны команды для загрузки соответствующих библиотек.

- d. Нужно иметь в виду, что приложения будут тестироваться на Ubuntu 20.04.

### **Дополнительные баллы:**

Эти требования можно не выполнять, но они пойдут плюсом, если будут реализованы.

1. Добавить возможность к каждому слову добавить соответствующую картинку. Картинки помогают запоминать слова. Реализовать можно следующим образом:
  - a. добавить третий столбец с названием «Image». В этом столбце указать имя файла, которое соответствует данному слову;
  - b. собрать все файлы в zip архив;
  - c. на странице загрузки словаря добавить еще одно поле загрузки файла, в который мы будем указывать zip-архив с картинками. Приложение должно сохранить картинки на сервере;
  - d. на главном экране приложения можно отобразить изображение, связанное с словом.
2. Добавить возможность прослушать, как звучит отображаемое слово и его перевод. Для реализации этого функционала можете использовать «Web Speech API» или сторонние сервисы.
3. Реализовать дополнительную страницу «Словарь», на которой вывести все слова из словарей и организовать поиск по словам.