

## **Лабораторная работа №12**

### **Изучение фреймворка Vue.js**

**Цель:** изучить основы фреймворка Vue.js и приобрести навыки создания простейшей веб-страницы с реактивным поведением.

#### **Порядок выполнения лабораторной работы:**

1. Подготовка к выполнению лабораторной работы – изучение основ Vue.js:
  - a. Знакомство с основными концепциями: конструктор, компоненты, директивы, переходы.
  - b. Реактивность данных и способы отслеживания.
  - c. Подключение фреймворка Vue.js.
  - d. Создание экземпляра Vue.js.
  - e. Работа с условиями, циклами, событиями и формами.
2. Изучить материал по Vue.js в руководстве.
3. Создание GIT репозитория для проекта, поэтапное сохранение результатов работы в репозитории.
4. Разработка и верстка страницы с использованием фреймворка Vue.js.
5. Размещение страницы на сервере факультета fit.mospolytech.ru

#### **Содержание и требования к веб-странице**

На странице должны присутствовать следующие элементы:

1. Заголовок первого уровня (h1)
2. Абзац с кратким описанием работы (p)
3. Заголовки второго уровня (h2)
4. Правильно структурированная форма с полями, подписями к полям и кнопкой
5. Счетчик символов
6. Сообщение о пустом списке
7. Список, когда есть данные для отображения

#### **Оформление:**

1. Подключен шрифт из библиотеки Google Fonts
2. Заданы отступы, таким образом чтобы блоки были визуально отделены друг от друга
3. Задан фон
4. Изменены стандартные стили полей и кнопки

#### **Логика работы страницы:**

Пока не заполнены оба поля, кнопка неактивна. Под полями находится счетчик знаков, он отображает сумму количества символов, введенных в оба поля. Если нет данных для отображения списком выводится сообщение «Список пуст». Примерный вид страницы показан на рисунке 1.

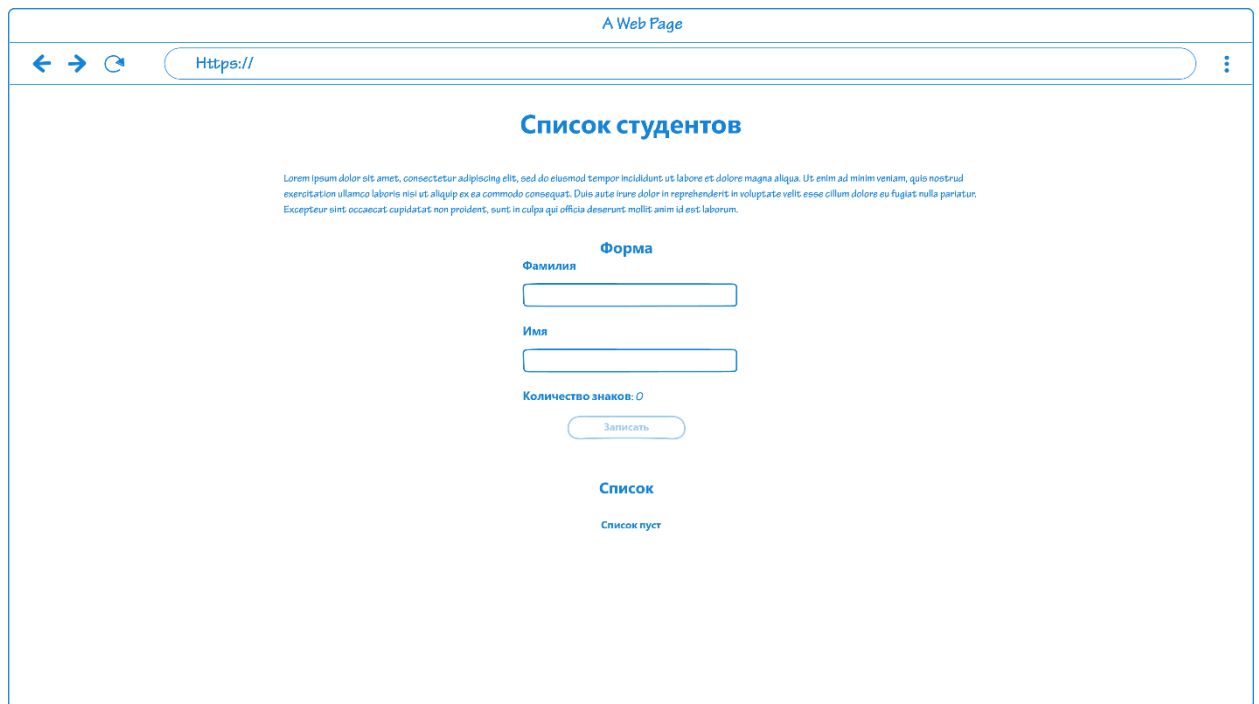


Рисунок 1 - Общий вид страницы сразу после загрузки

После заполнения обоих полей, кнопка становится активной и при ее нажатии имя и фамилия записываются и отображаются в списке, из полей стираются данные. При повторении действий список пополняется (рис. 2).

Перед записью фамилии и имени строки обрабатываются: первый символ должен быть заглавным, остальные строчные. Например, если пользователь ввел иваноВ, то записаться должно Иванов.

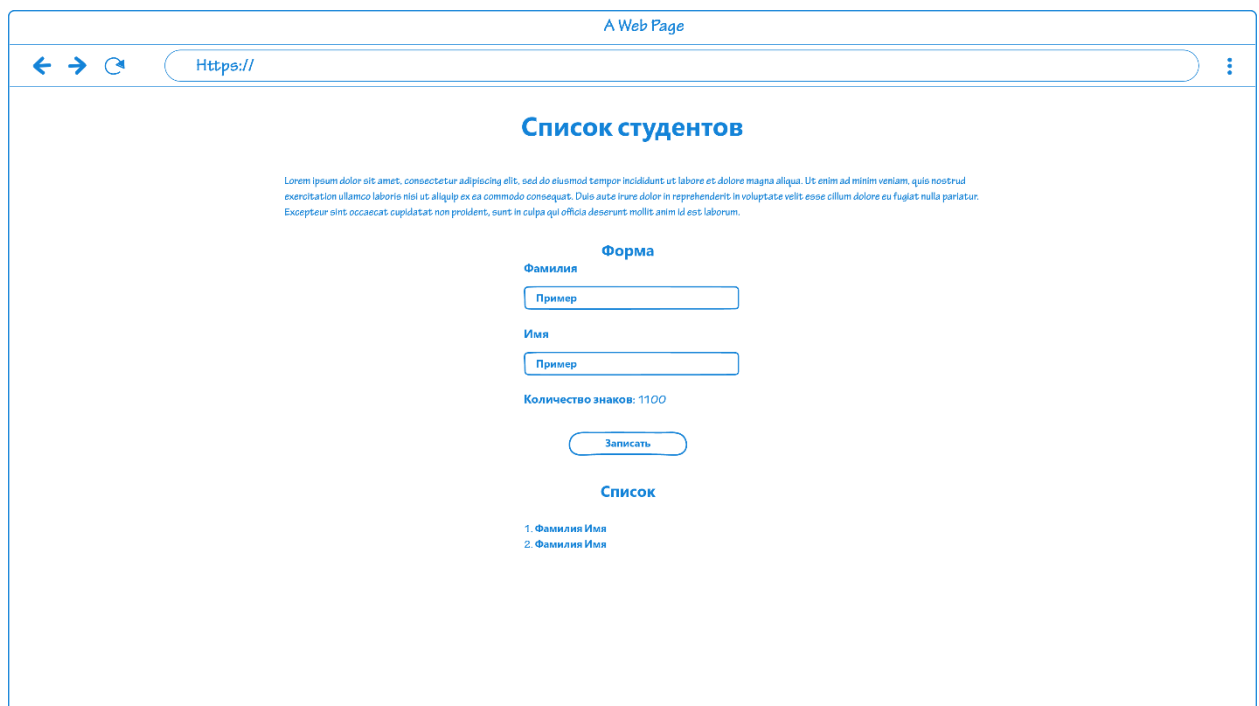


Рисунок 2 - Общий вид страницы после внесения данных

#### Действия:

1. Создать новый экземпляр Vue.js.
2. Связать элементы формы и состояния экземпляра с помощью директивы v-model.
3. Добавить обработку событий v-on на кнопку.
4. Для управления атрибутом кнопки использовать v-bind.
5. Отображать и скрывать сообщение «Список пуст» через v-show или v-if.
6. С помощью v-for создать динамическое отображение заполняемого списка.
7. Вычисляемое свойство задействовать для подсчета количества знаков.

#### Требования:

1. Веб-страница содержит все требуемые элементы и оформление.
2. Веб-страница работает по описанной в задании логике.
3. Отсутствуют ошибки на валидаторе.
4. Отсутствуют ошибки в консоли браузера при работе с веб-страницей.

#### Результаты выполнения лабораторной работы:

Веб-страница, удовлетворяющая требованиям, размещенная в личном кабинете на сервере факультета fit.mospolytech.ru и на удаленном репозитории с историей коммитов.

#### Основные и дополнительные источники:

1. Руководство Vue.js на русском - <https://ru.vuejs.org/v2/guide/>
2. «Поваренная книга» по Vue.js - <https://ru.vuejs.org/v2/cookbook/>
3. Хэнчеп Эрик, Листуон Бенджамин «Vue.js в действии», 2019.
4. Статья «Переход с jQuery на Vue.js» - <https://habr.com/ru/company/ruvds/blog/414073/>
5. Короткий курс на английском языке - <https://www.vuemastery.com/courses/intro-to-vue-js/vue-instance>