Практическая работа 1.2

Web-приложения в Node.js

### Цель работы

Данная практическая работа направлена на отработку и закрепление навыков создания примитивных web-приложений с использованием фреймворка express.js и шаблонизатора liquid.

Задание

В рамках выполнения практической работы требуется выполнить следующие работы и продемонстрировать полученные навыки:

* Создать новое node.js приложение;
* Подключить к приложению библиотеки express.js и liquid;
* Разработать HTML-шаблоны для страниц приложения
* Реализовать логику обработки на сервере согласно варианту задания

Содержание отчета

* Цель работы
* Условие задачи, которая будет решаться
* Скриншоты получившегося приложения
* Исходный код решения

Задачи

Задача 1

Необходимо реализовать простой калькулятор, выполняющий сложение, вычитание, умножение и деление двух чисел. На форме калькулятора располагаются элементы управления для ввода первого операнда, операции и второго операнда. По нажатию на кнопку «Вычислить» отображается результат операции. Вычисление должно выполняться на сервере. Выполненные операции должны записываться в файл на сервере и отображаться пользователю на отдельной странице.

Задача 2

Необходимо реализовать простой калькулятор, выполняющий сложение, вычитание, умножение и деление двух чисел. На форме приложения отображается классическая клавиатура калькулятора и его экран в виде поля текстового ввода. Нажатие на цифровые клавиши добавляет новую цифру на экран, знаки операций и клавиша = выполняет вычисления на сервере. Выполненные операции должны записываться в файл на сервере и отображаться пользователю на отдельной странице.

#### Задача 3

То же самое, что и в задаче № 2, но все клавиши калькулятора работают на сервере.

#### Задача 4

Реализовать примитивную систему тестирования, записывающую ответы заранее прописанные в системе вопросы. При открытии приложения пользователю отображается форма, где можно ввести имя, а также ответы на вопросы. Ответы пользователей записываются в файл. На отдельной странице реализовать таблицу, отображающую количество правильных ответов для пользователей, которые прошли тест.

#### Задача 5

То же самое, что и в задаче № 4, но вопросы и ответы также заданы текстовых в файлах на сервере.

#### Задача 6

Реализовать простейший конфигуратор автомобилей (или каких-нибудь других вещей, которые могут составляться из нескольких опций). При открытии приложения пользователю отображаются элементы управления, позволяющие выбрать марку автомобиля, двигатель, пакеты опций. При переключении элементов управления должна обновляться цена, которая также отображается на форме. Вычисление цены должно происходить на сервере. Также на форме располагается поле для ввода адреса электронной почты пользователя и кнопка Сохранить. При нажатии на эту кнопку, выбранные пользователем опции, а также его email сохраняются в файл на сервере.

#### Задача 7

То же самое, что и № 6, но называния и варианты дополнительных опций, а также их цены задаются в текстовых файлах на сервере.