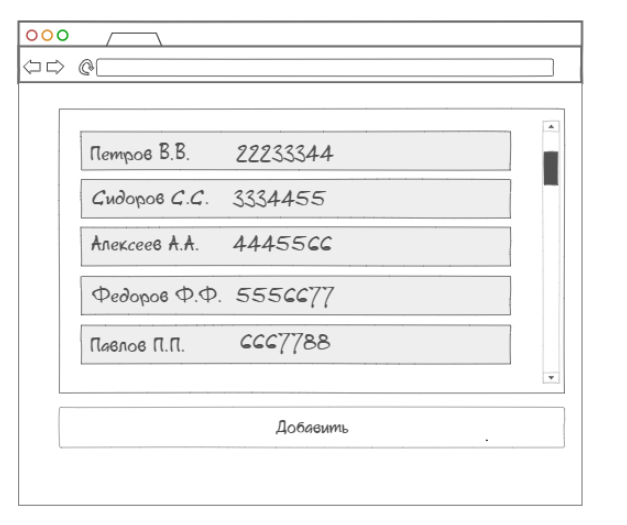
Лабораторная работа 15

ПСКП

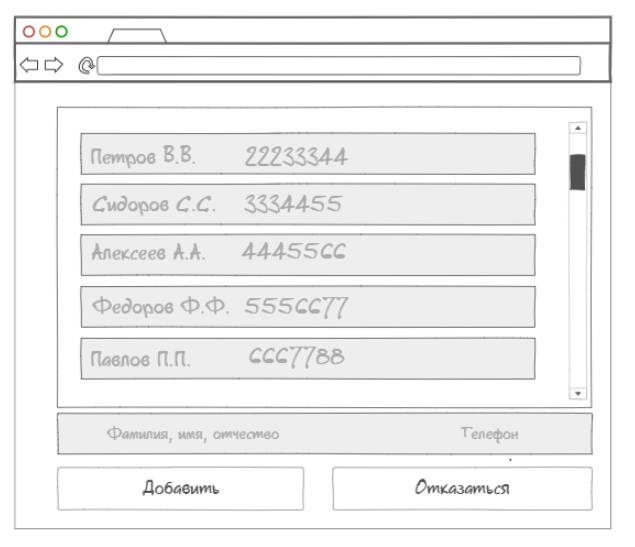
ПОИТ-3

**Задание 01**

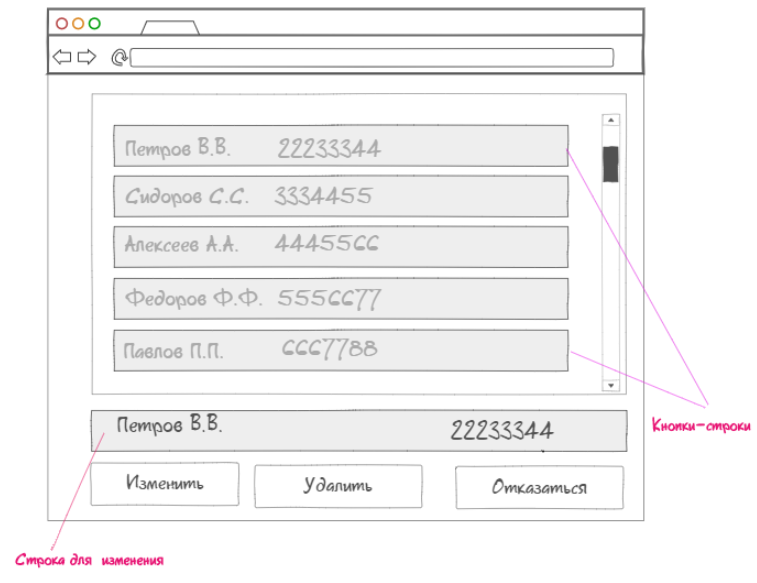
1. Разработайте приложение **15-01**, представляющее собой EXPRESS-HTTP-сервер, прослушивающий порт ***3000***.
2. Приложение **15-01** предназначено для поддержки телефонного справочника.
3. Данные справочника располагаются в json-файле, json-структуру документа разработайте самостоятельно. json-файл должен располагаться в корневой директории приложения.
4. Запрос **GET:/** должен отображать в окне браузера следующую форму.

**GET:/**

1. Форма **GET:/** позволяет добавлять новую строку справочника (кнопка **Добавить**) или выбирать строку из списка (нажать на **строку-кнопку**).
2. Нажатие на **строку-кнопку** на форме **GET:/** отправляет запрос **GET:/Update.**
3. Нажатие кнопки **Добавить** на форме **GET:/** отправляет запрос **GET:/Add**, который отображает в окне браузера следующую форму.

**GET:/Add**

1. Форма **GET:/Add** позволяет ввести строку справочника (ввод данных строки и нажать кнопку **Добавить**) или отказаться (нажать на кнопку **Отказаться**).
2. Нажатие кнопки **Добавить** на форме **GET:/Add** приводит к добавлению строки в json-файл (запрос **POST:/Add**) и отображение формы **GET:/** вокне браузера.
3. Нажатие кнопки **Отказаться** на форме **GET:/Add** приводит к отправке запроса **GET:/**.
4. **Кнопки-строки** на форме **GET:/Add** должны быть заблокированы.
5. Запрос **GET:/Update** (при нажатии **строки-кнопки** на форме **GET:/**) приводит к отображению следующей формы.

**GET:/Update**

1. На форме **GET:/Update кнопки-строки** заблокированы, в **строке для изменения** находятся данные соответствующей **кнопки-строки** на форме **GET:/**.
2. **Строка для изменения** позволяет ввести изменения выбранной строки в json-файл с помощью кнопки **Изменить** (запрос **POST:/Update**)**,** после изменения в браузере отображается форма **GET:/**. При вводе первого символа в **строку для изменения** кнопка **Удалить** должна быть заблокирована (js-код).
3. Кнопка **Удалить** на форме **GET:/Update** удаляет строку в json-файле (запрос **POST:/Delete**) и отображает форму **GET:/**.
4. Кнопка **Отказаться** на форме **GET:/Update** выполняет запрос **GET:/.**
5. Все http-запросы сведены в следующей таблице.

|  |  |
| --- | --- |
| **GET:/** | Отображается форма **GET:/** |
| **GET:/Add** | Отображается форма **GET:/Add** |
| **GET:/Update** | Отображается форма **GET:/Update** |
| **POST:/Add** | Добавляет строку в json-файл, в ответе форма **GET:/** |
| **POST:/Update** | Изменяет строку в json-файле, в ответе форма **GET:/** |
| **POST:/Delete** | Удаляет строку в json-файле, в ответе форма **GET:/** |

1. Описание форм приведено в следующей таблице.

|  |  |
| --- | --- |
| **GET:/** | теги <а> для всех кнопок |
| **GET:/Add** | теги <а> для всех **строк-кнопок**, кнопка **Oтказаться**, теги <form/input/button> для **строки для изменения** икнопки **Добавить** |
| **GET:/Update** | теги <а> для всех **строк-кнопок**, кнопка **Oтказаться**, теги <form/input/button> для **строки для изменения** икнопки **Изменить** и **Удалить** |

1. Приложение **15-01** должно использовать пакет **EXPRESS**.
2. Приложение **15-01** должно использовать пакет **EXPRESS-HANDLEBARS**.
3. Приложение **15-01** должно использовать механизм статических файлов (css, js) пакета **EXPRESS**.
4. Все представления должны быть реализованы с помощью handlebars-template.
5. В приложении **15-01** все представления должны использовать общий макет.
6. При реализации **кнопок-строк** всех представлений используйте общее **partial-view**.
7. При реализации кнопки **Отказаться** используйте **helper**.
8. Тестирование выполнить с помощью браузера.

**Задание 02**

1. Разместите приложение **15-01** на Render (<https://render.com/>). Подробнее про развертывание: <https://virendraoswal.com/nodejs-deployment-via-rendercom>
2. Тестирование выполнить с помощью браузера.

**Задание 03** Ответьте на следующие вопросы

1. Поясните понятие **Шаблонизация**.
2. Поясните понятие **Шаблон (template).**
3. Поясните понятие **Контекст шаблона (template context).**
4. Поясните понятие **Макет (layout).**
5. Поясните понятие **Частичное представление (partial-view).**
6. Поясните понятие **Хелпер** (**helper)**.
7. Перечислите встроенные хелперы Handlebars.
8. Поясните понятие **Статический файл**.
9. Поясните назначение http-заголовков: **Cache-control, ETag, Last-Modified**, **If**-**Modified-Since, If-None-Match, Expires.**
10. Поясните свойство **\_\_dirname** объекта **global**.
11. Поясните назначение и принцип работы метода **response.download().**
12. Поясните назначение и принцип работы метода **response.attachment().**
13. Поясните назначение http-заголовка: **Content-Disposition: attachment.**
14. **Шаблонизация** - это процесс создания документов или кода, в которых части содержимого (обычно текстовые строки) заменяются значениями из некоторых источников данных. Это позволяет создавать динамические контентные элементы, такие как веб-страницы, электронные письма или документы, используя шаблоны с заполнителями, которые заменяются реальными данными при генерации.
15. **Шаблон (template)** - это структура, определенная для создания документов или кода, которая содержит заполнители или метки, которые должны быть заменены на конкретные значения при генерации контента. Шаблон обычно представляет собой текстовый файл с разметкой и специальными символами для обозначения заполнителей.
16. **Контекст шаблона (template context)** - это объект, который содержит данные, используемые для заполнения шаблона. Контекст шаблона обычно представляет собой объект JavaScript, который содержит переменные и их значения, которые будут использоваться в шаблоне.
17. **Макет (layout)** - это общая структура или каркас, который определяет общий вид и оформление страницы или другого документа. Он содержит общие элементы дизайна, такие как заголовок, навигационное меню, подвал и т. д., которые могут быть использованы на нескольких страницах. В шаблонизации веб-приложений макеты часто используются для создания единообразного внешнего вида для всех страниц.
18. **Частичное представление (partial-view)** - это маленький фрагмент шаблона, который может быть включен в другие шаблоны или макеты. Частичные представления используются для повторного использования кода и создания более модульной структуры шаблонов.
19. **Хелпер (helper)** - это функция или метод, предназначенный для выполнения определенной задачи в шаблоне. Он может быть использован для обработки данных перед их выводом, выполнения операций сравнения, форматирования или других манипуляций с данными во время генерации контента.
20. **Встроенные хелперы Handlebars**:
    * **if**: условный оператор.
    * **each**: цикл для итерации по массивам или объектам.
    * **unless**: отрицание условия.
    * **with**: изменение контекста.
    * **log**: вывод отладочной информации.
21. **Статический файл** - это файл, который не изменяется динамически во время выполнения программы или веб-приложения. Обычно это файлы таких типов, как изображения, CSS, JavaScript, шрифты и другие файлы, которые не требуют обработки на стороне сервера перед отправкой клиенту.
22. **HTTP-заголовки**:
    * **Cache-control**: управляет кэшированием ресурса на стороне клиента или прокси-сервера.
    * **ETag**: тег сущности, используемый для определения изменений в содержимом ресурса.
    * **Last-Modified**: дата и время последней модификации ресурса.
    * **If-Modified-Since**: условное заголовок, используемый клиентом для проверки изменений в ресурсе с момента его последнего обновления.
    * **If-None-Match**: условный заголовок, используемый клиентом для проверки ETag ресурса.
    * **Expires**: указывает дату и время, когда ресурс станет недействительным и должен быть перезапрошен с сервера.
23. Свойство **\_\_dirname** объекта **global** представляет путь к текущему каталогу скрипта Node.js.
24. Метод **response.download()** используется для отправки файла как вложения браузеру пользователя. Он указывает браузеру загрузить файл с указанным именем файла.
25. Метод **response.attachment()** устанавливает заголовок Content-Disposition на **attachment**, что указывает браузеру открывать вложение в новом окне или скачивать его, а не отображать его в браузере.
26. HTTP-заголовок **Content-Disposition: attachment** сообщает браузеру, что в ответе содержится вложение, которое должно быть загружено или сохранено, а не отображено в браузере.