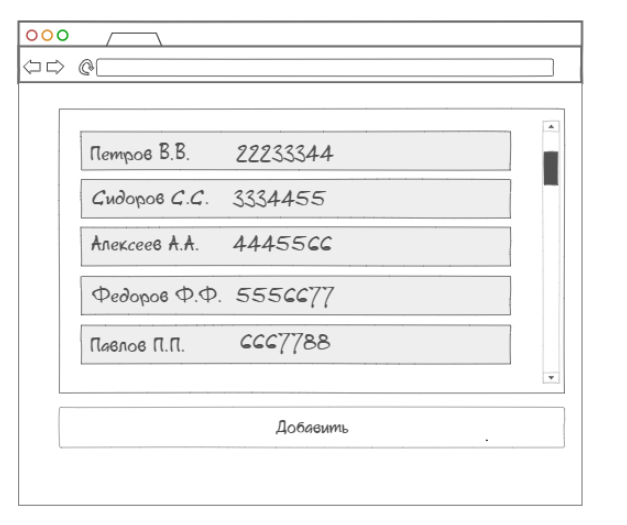
Лабораторная работа 20

ПСКП

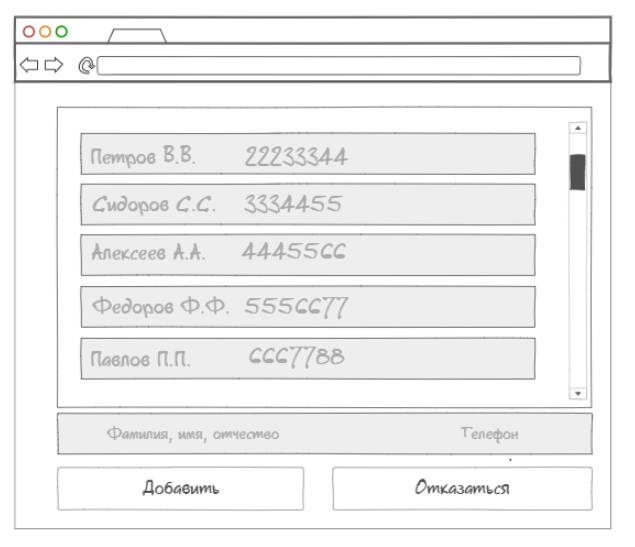
ПОИТ-3

**Задание 01**

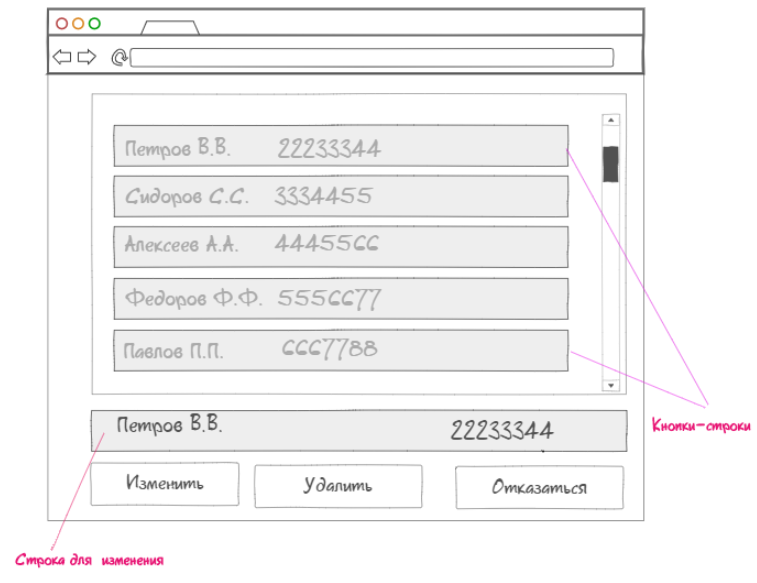
1. Разработайте приложение **20-01**, представляющее собой EXPRESS-HTTP-сервер, прослушивающий порт ***3000***.
2. Приложение **20-01** предназначено для поддержки телефонного справочника.
3. Данные справочника располагаются в json-файле, json-структуру документа разработайте самостоятельно. json-файл должен располагаться в корневом 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 для из списка (нажать на **строку-кнопку**).
2. Нажатие на **строку-кнопку** на форме **GET:/** оправляет 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** находится данные соответствующей **кнопке-строке** 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 сведено в следующей таблице.

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

1. Приложение **20-01** должно использовать модуль **EXPRESS**.
2. Приложение **20-01** должно использовать модуль **EXPRESS-HANDELBARS**.
3. Приложение **20-01** должно использовать механизм статических файлов (css, js) модуля **EXPRESS**.
4. Все формы должны быть реализованы с помощью handelbars-template.
5. В приложении **20-01** все формы должны использовать общий макет.
6. При реализации **кнопок-строк** всех форм используйте общее **partial-view**.
7. При реализации кнопки **Отказаться** используйте **helper**.
8. Тестирование выполнить с помощью браузера.

**Задание 03**

1. Разместите приложение **20-01** на платформе HEROKU.
2. Тестирование выполнить с помощью браузера.

**Задание 04** 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 Handelbars.
8. Поясните понятие **Статический файл**.
9. Поясните назначение http-заголовков: **Cache-control, ETag, Last-Modified**, ***If***-**Modified-Since, If-None-Match.**
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-заголовков: **Content-Disposition: attachment**