|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 3 |

**Название:**

Работа с POST и GET запросами, шаблонизаторами и cookie

**Дисциплина:** Архитектура ЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-52Б |  |  | Брянская Е.В. |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Попов А.Ю. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

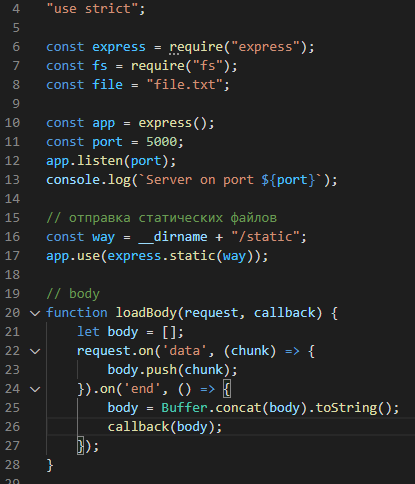
**Цель:** научиться работать с POST и GET запросами, корректно взаимодействовать с сервером, также изучить cookie. Выполнить соответствующие задания на основе изученного материала.

**Часть 1**

**Задание 1**

Создать сервер. Сервер должен выдавать страницу с тремя текстовыми полями и кнопкой. В поля ввода вбивается информация о почте, фамилии и номере телефона человека. При нажатии на кнопку "Отправить" введённая информация должна отправляться с помощью **POST** запроса на сервер и добавляться к концу файла (в файле накапливается информация). При этом **на стороне сервера** должна происходить проверка: являются ли почта и телефон уникальными. Если они уникальны, то идёт добавление информации в файл. В противном случае добавление не происходит. При отправке ответа с сервера клиенту должно приходить сообщение с информацией о результате добавления (добавилось или не добавилось). Результат операции должен отображаться на странице.

**index.js**

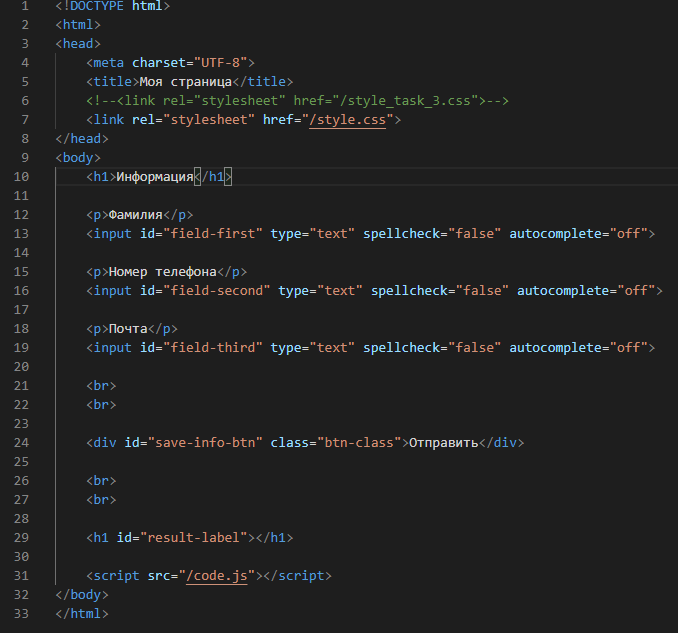




**code.js**

****

**page.html**

****

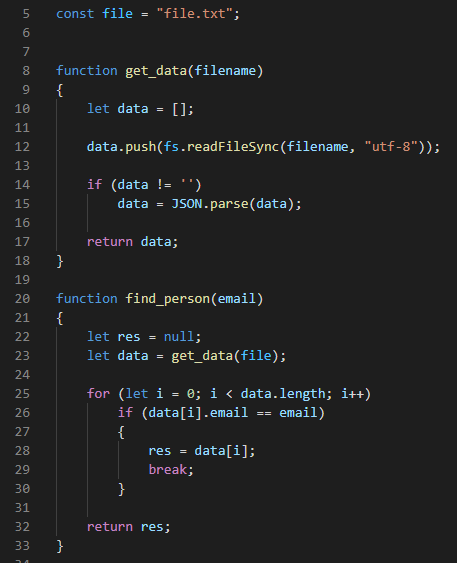
**Тесты:**

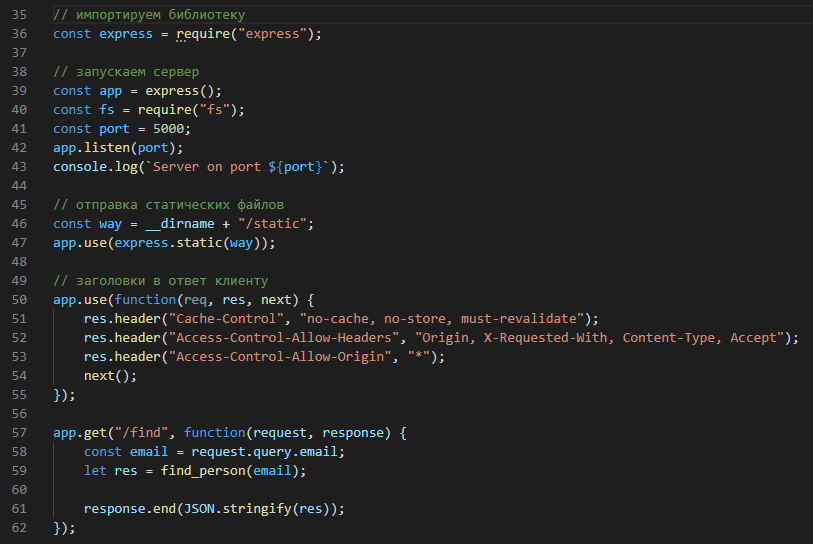
|  |  |
| --- | --- |
| Тест | Результат |
| Файл пустой |  |
| В файле содержится информация с прошлого запроса |  |
|  |  |

**Задание 2**

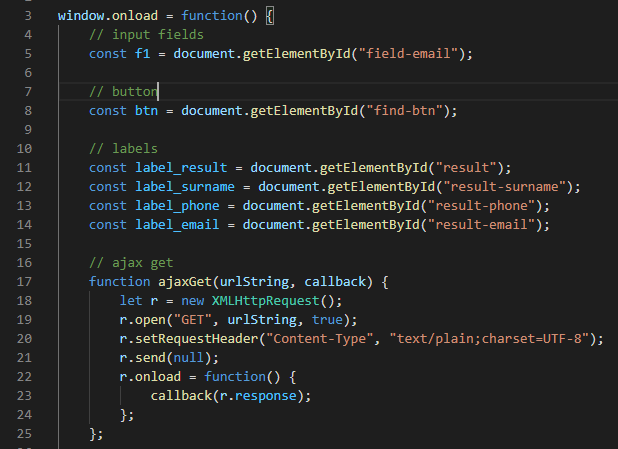
Добавить серверу возможность отправлять клиенту ещё одну страницу. На данной странице должно быть поле ввода и кнопка. В поле ввода вводится почта человека. При нажатии на кнопку "Отправить" на сервер отправляется **GET** запрос. Сервер в ответ на **GET** запрос должен отправить информацию о человеке с данной почтой в формате **JSON** или сообщение об отсутствии человека с данной почтой.

**index.js**

****

****

**code.js**

****

****

**page.html**

****

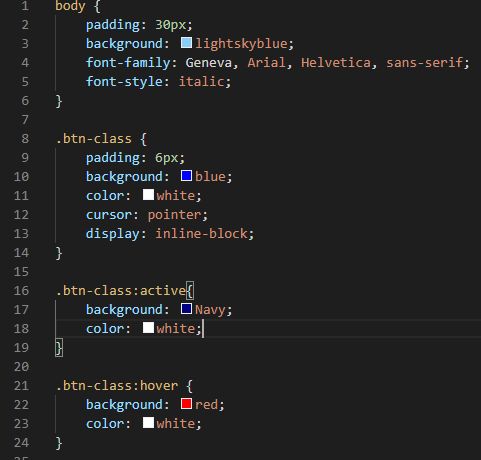
**Тесты:**

|  |  |
| --- | --- |
| Тест | Результат |
| Пользователь существует |  |
| Пользователя нет |  |

**Задание 3**

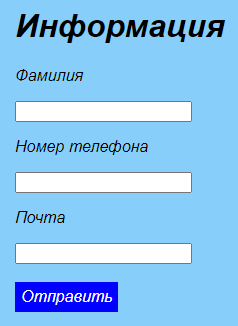
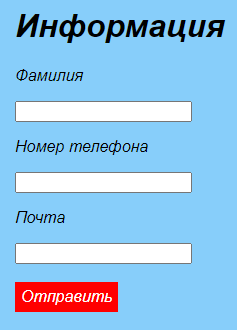
Оформить внешний вид созданных страниц с помощью **CSS**. Информация со стилями **CSS** для каждой страницы должна храниться в отдельном файле. Стили **CSS** должны быть подключены к страницам.

**style.css** (для первого задания)

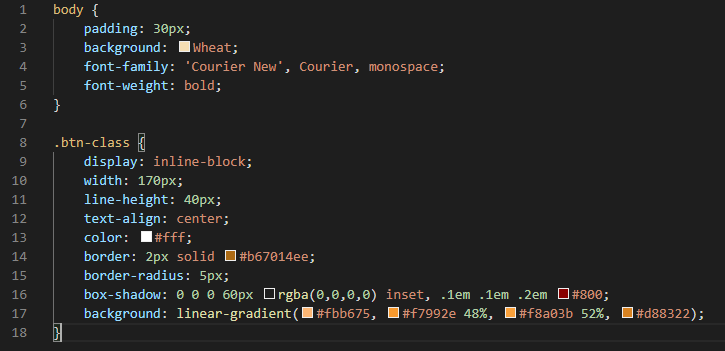


**Результат:**

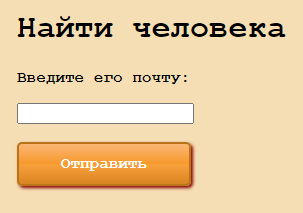
(при наведении на кнопку происходит смена цвета)

**style.css** (для второго задания)



**Результат:**

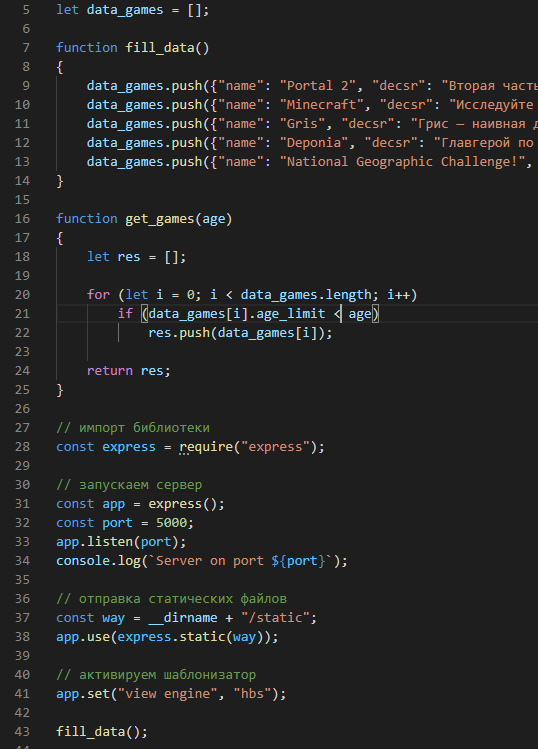


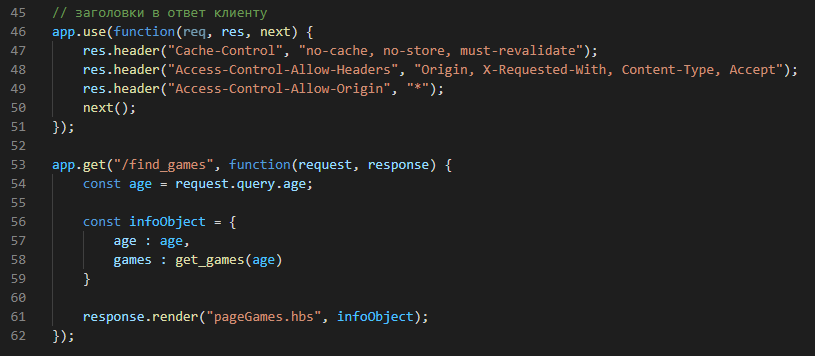
**Часть 2**

**Задание 1**

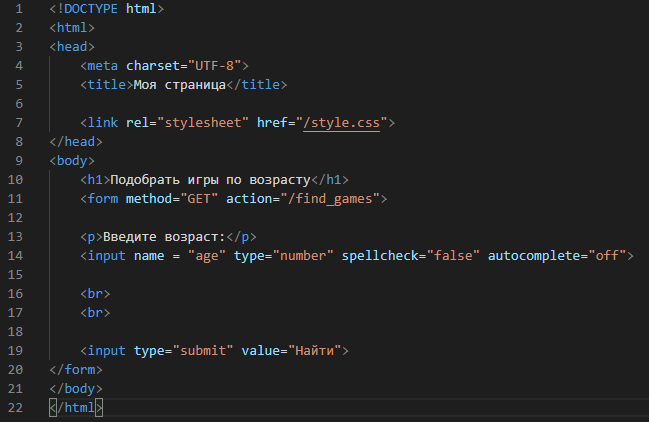
Создать сервер. В оперативной памяти на стороне сервера создать массив, в котором хранится информация о компьютерных играх (название игры, описание игры, возрастные ограничения). Создать страницу с помощью шаблонизатора. В **url** передаётся параметр возраст (целое число). Необходимо отображать на этой странице только те игры, у которых возрастное ограничение меньше, чем переданное в **url** значение.

**index.js**

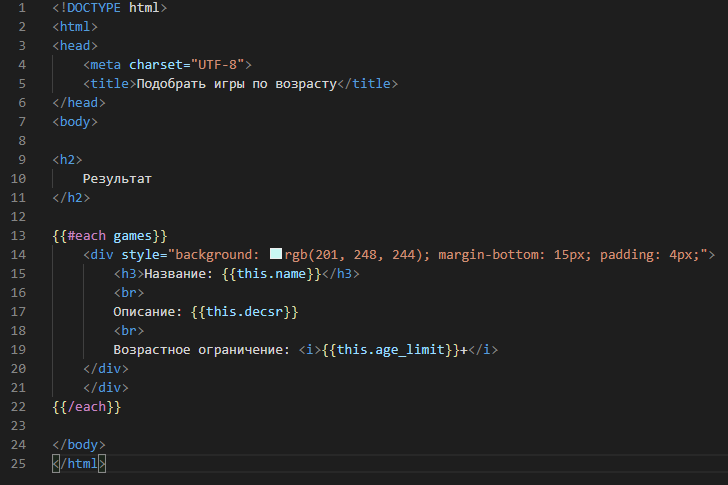
****

****

**Page.html**

****

**pageGames.hbs**

****

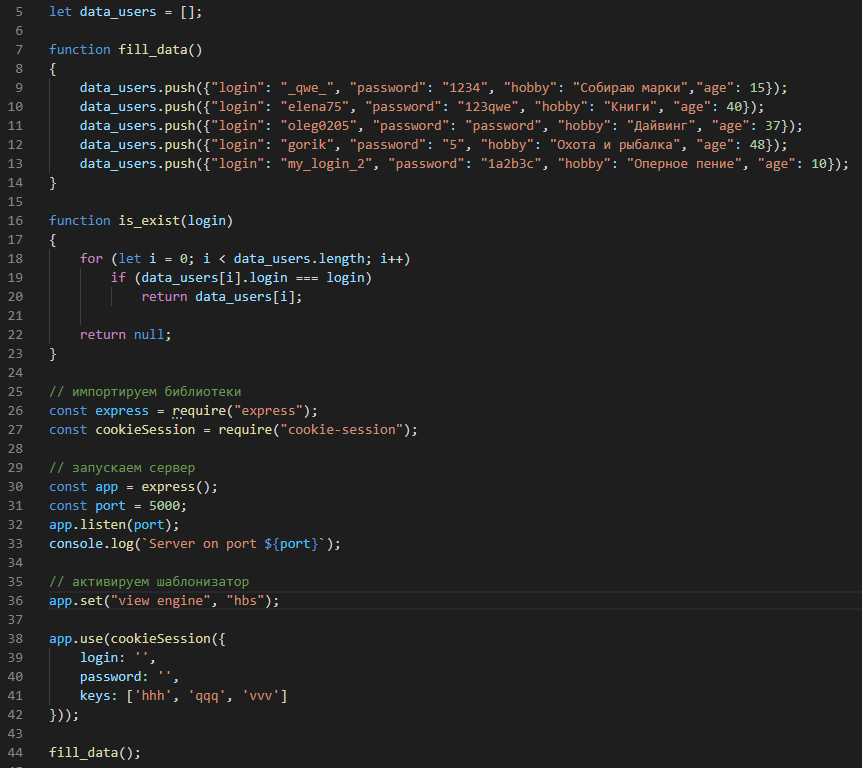
**Тесты:**

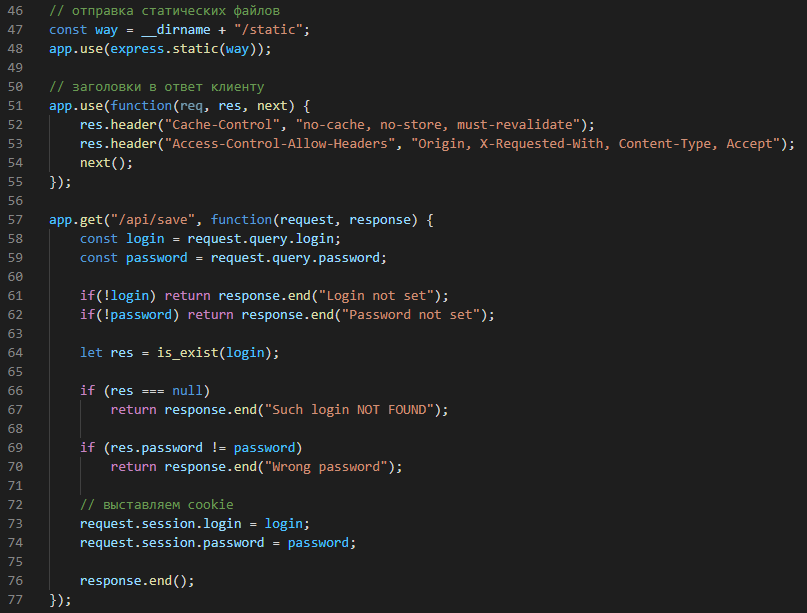
|  |  |
| --- | --- |
| Тест | Результат |
|  |  |
|  |  |

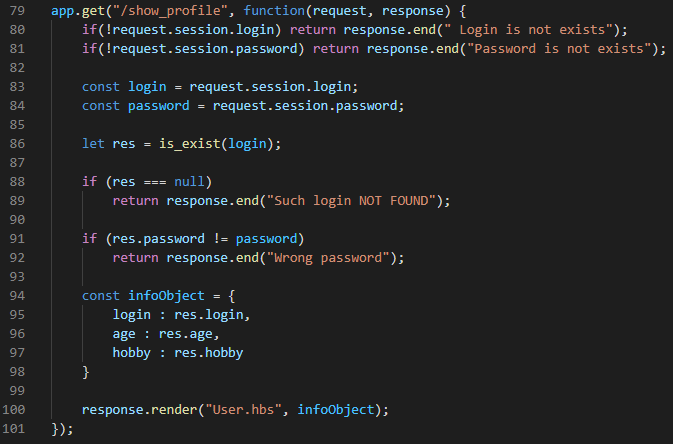
**Задание 2**

Создать сервер. В оперативной памяти на стороне сервера создать массив, в котором хранится информация о пользователях (логин, пароль, хобби, возраст). На основе **cookie** реализовать авторизацию пользователей. Реализовать возможность для авторизованного пользователя просматривать информацию о себе.

**index.js**



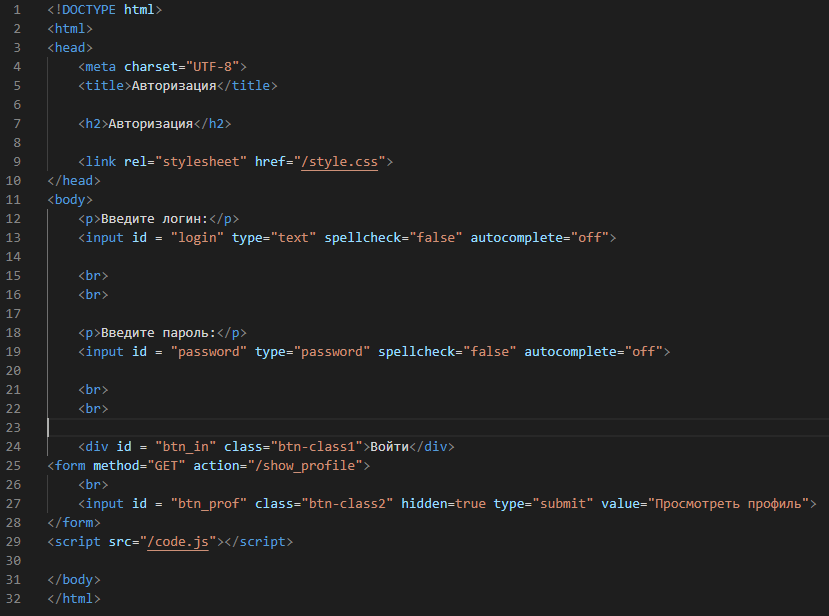




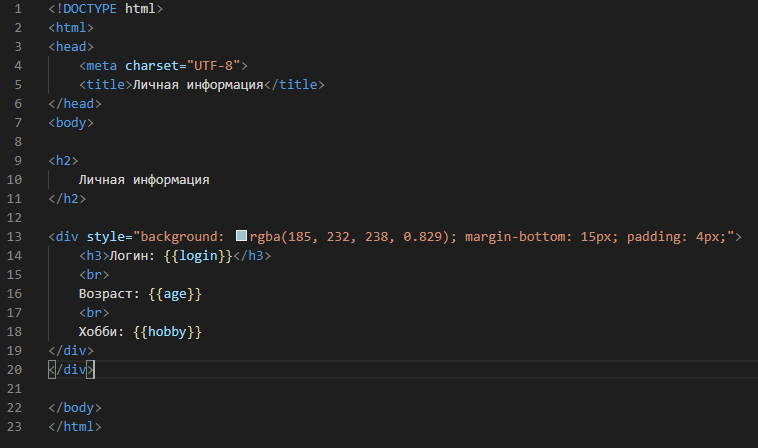
**code.js**



**page.html**



**User.hbs**



**Тесты:**

|  |  |
| --- | --- |
| Тест | Результат |
| Пользователь существует |  |
| Пользователя нет |  |

**Вывод**

В ходе лабораторной работы были изучены POST и GET запросы, шаблонизаторы, методы работы с cookie. На основе полученных знаний были выполнены задания лабораторной работы, результаты представлены в виде отчёта.