OK Stop. Хакатон. Задания до 14.04 (должны быть готовы в вечер пятницы 13.04)

Лёша:

1. Html
   1. **Логин-страница** (поля для ввода + кнопка)
   2. **Страница ввода** (таблица + кнопка «сформировать», см. рисунок в блокноте «ФТШ»)  
      Остальные поля можно пока не делать либо оставить нерабочими
   3. **Страница вывода** (пока только таблица)
   4. **ПОТОМ!!!** Страница со списком доступных эвентов и фильтром и поиском по ним, появляющаяся по нажатию кнопки (совсем потом! Сначала всё остальное! С этим будут связаны задания Валере (запросы к БД) и Мише и Никите (написать фильтр и поиск))
2. Js-функции
   1. **Логин-страница**: на клик кнопки запуск функций от Валеры get\_group\_id (login, pass)  
      Если вернулось -1, очищает поля ввода и просить ввести ещё раз пароль и логин

Если вернулось id, запускает функцию от Валеры redirect\_to\_input\_page(ID)

* 1. **Страница ввода**:

login=getCookie('login');

pass=getCookie('pass');

if(login==-1||pass==-1)

window.alert("пожалуйста, авторизуйтесь");

(или что-нибудь такое происходит, что просит автроизоваться)Можно оформить в check\_if\_logged\_in() (собственно, эту функцию я уже написал)  
events\_id=get\_raw\_events\_id(login,pass,0), где 0 – номер дня – пишет Валера  
Тебе нужно написать: функция raw\_events\_table(events\_id) выводит таблицу имён (пока просто имён) событий в html, пользуясь функциями get\_online(login,pass,online\_id),get\_lesson\_name(lesson\_id,group\_id), get\_elective\_name(elective\_id,day), кот. пишет Валера  
Важно: events\_id – одномерный массив, все события пользователя в данный день  
ev\_id – одна переменная, событие, которому в БД надо найти имя  
Ещё тебе - функция, кот. по нажатию «сформировать» вызывает три функции:   
calc(login,pass,day) (пишут Миша и Никита)  
redirect\_to\_output\_page(login,pass) (пишет Валера)  
write\_raw\_events\_id(login,pass, events\_id,day) (Валера)

* 1. **Страница вывода:**login=getCookie('login');

pass=getCookie('pass');

if(login==-1||pass==-1)

window.alert("пожалуйста, авторизуйтесь");

(или что-нибудь такое происходит, что просит автроизоваться)

res=get\_schedule(usr\_id, 0) – возвращает из БД двумерный массив, результат работы calc. Пишет Валера  
**Тебе нужно:** функция events\_table(res), которая рисует в html табличку

**Миша и Никита:**

Функция   
calc(login,pass,day){

events\_id=[201, 202, 203, 204]; //это данные для тестирования  
 //events\_id=get\_raw\_events\_id(login,pass,day);  
 res\_long=calc\_long(events\_id,login,pass); // Миша  
 res\_short=calc\_short(res\_long,login,pass); // Никита  
 return res\_short;

}

и, соответственно, функции calc\_long/short

Миша:

Базовый алгоритм работы calc\_long:

1. Расставляет пары
2. Добавляет в свободное время онлайн-курсы, исходя из того, какое минимальное время работы возможно для этого задания
3. Возвращает день в виде одномерного массива пятиминуток. В каждой пятиминутке - ID события либо -1

ЭТИ ДВА ПУНКТА ЖЕЛЕЗНО ДОЛЖНЫ БЫТЬ НАПИСАНЫ И РАБОТАТЬ КАК ЧАСЫ

Предполагается использование функций (описание функций - в разделе “Валера”):

get\_user\_info(login, pass)  
get\_lesson\_start(lesson\_id, group\_id, day)  
get\_lesson\_stop(lesson\_id, group\_id, day)  
get\_nearest\_online(login, pass)

В местах, где требуется при помощи этих функций запрашивать данные из БД, код пишется, например, так:  
var user\_info = [“valerii”, “zuev”, “136322”,”1jU6#fAlg5ie”]  
//var user\_info=get\_user\_info(login, pass);

Так можно проверить работу на тестовых данных, не обращаясь к БД. Когда я читаю этот код, я вижу закомментированную строку и, если у меня уже есть написанная функция get\_user\_info, снимаю комментирование и тестирую в боевых условиях

Далее нужно добавить заполнение времени доп. занятиями (см. часть работы, за которую ответственный Валера)  
Здесь будет полезна функция get\_elective\_properties(elective\_id,day)

Сначала calcs работает только в предположении, что события с фиксированным временем не пересекаются. В дальнейшем следует добавить в calc\_long  
function calc(events\_id, usr\_id){  
 //запрашивает данные из БД…

//добавляет расписание…  
 //Пытается добавить ещё что-то с фиксированным временем и видит, что время занято.

Var Error\_log=[];

Error\_log.push(-1);

Error\_log\_push(конфликтное событие 1)

Error\_log\_push(конфликтное событие 2)

return(error\_logs)

}  
Далее следует предупредить пользователя и попросить его убрать одно из событий

Никита:

Calc\_short из массива пятиминуток составляет двумерный массив - таблицу, которая содержит (пока) события в хронологическом порядке в такой форме:

|  |  |  |
| --- | --- | --- |
| Имя события | Начало события | Конец события |
| Лекция по матанализу | 12,00 | 13,40 |

Функции, которыми при этом предполагается пользоваться:

get\_online(login, pass, online\_id)  
get\_lesson\_name(lesson\_id,group\_id)  
get\_lesson\_start(lesson\_id, group\_id, day)  
get\_lesson\_stop(lesson\_id, group\_id, day)  
write\_schedule(schedule,day);  
get\_elective\_name(elective\_id,day) (это уже в расширенной версии - см. задание для calc\_long)  
get\_elective\_properties(elective\_id,day)

Валера:

Задачи - написать js-файл db\_calls.js со след. функциями:

get\_group\_name(login, pass)

redirect\_to\_input\_page(login,pass)

getCookie('login');

getCookie('pass');  
get\_raw\_events\_id(login,pass,day)  
write\_raw\_events\_id(login,pass, events\_id,day)  
get\_schedule(login,pass, day)

get\_user\_info(login, pass)  
get\_lesson\_start(lesson\_id, group\_id, day)  
get\_lesson\_stop(lesson\_id, group\_id, day)  
get\_nearest\_online(login, pass)  
get\_online(login, pass,online\_id)  
get\_lesson\_name(lesson\_id,group\_id,day)  
write\_schedule(schedule,day)

В дальнейшем:   
get\_elective\_properties(elective\_id,day)  
get\_elective\_name(elective\_id,day)

Описание функций:

Все функции, начинающиеся с get, обращаются к БД. Все, начинающиеся с write, записывают что-то в БД

get\_group\_id(login, pass)

login - строка  
 pass-строка  
 возвращает целое число – номер группы без разделителя, например 136322

redirect\_to\_input\_page(usr\_id)  
 usr\_id - строка  
 Ничего не возвращает. Выполняет переадресацию пользователя на страницу ввода в калькулятор, добавляя в URL суффикс - ID пользователя

getCookie('login');

getCookie('pass');  
 Каждая возвращает строку, если такая есть в файлах cookies браузера, либо -1  
get\_raw\_events\_id(login,pass,day)  
 login - строка  
 pass-строка  
 day - целое неотрицательное число. 0 - текущий день. 1 - завтра и т. д.  
 Возвращает одномерный массив целых чисел, каждое число от 1 до 410. Номера от 1 до 200 означают, что событие дополнительное. 201-210 - пары. 210 до 410 - онлайн-курс  
write\_raw\_events\_id(login,pass,events\_id,day)

login - строка  
 pass-строка  
 events\_id - одномерный массив   
 day - целое неотрицательное число. 0 - текущий день. 1 - завтра и т. д.   
 Записывает events\_id в БД. Возвращает 1, если успешно записано и -1 в ином случае.   
get\_schedule(login,pass, day)  
 login - строка  
 pass-строка  
 day - целое неотрицательное число. 0 - текущий день. 1 - завтра и т. д.  
 Выводит двумерный массив 3\*n, n от 0 до 20. В первом столбце имя события, во втором время начала, в третьем время конца.

get\_user\_info(login, pass)

Пока что выдаёт только одно число – номер группы. В дальнейшем предназначена для выдачи всякой личной информации типа настроек профиля, уведомлений и т. п.

get\_lesson\_start(lesson\_id, group\_id,day)

group\_id – целое число  
lesson\_id – целое число от 201 до 210  
day - целое неотрицательное число. 0 - текущий день. 1 - завтра и т. д.  
Возвращает целое число от 0 до 288 – номер пятиминутки с начала дня, когда начинается пара

get\_lesson\_stop(lesson\_id, group\_id, day) – аналогично предыдущей  
get\_nearest\_online(login, pass)

login - строка  
pass-строка  
Возвращает одномерный массив чисел из 4 эл-тов: первый – день, в который дедлайн по заданию (от 0 до +inf), второй – id задания, третий – минимальное время, которое имеет смысл занматься этим заданием, четвёртое – максимальное время, на которое расчитано всё задание

get\_online(login, pass,online\_id)

Возвращает одномерный массив строк. Первый элемент – имя события. Второй – минимальное время, которое стоит заниматься заданием. Третий – макс. Время

get\_lesson\_name(lesson\_id,group\_id, day)

lesson\_id – целое число от 201 до 210  
group\_id – целое число  
day - целое неотрицательное число. 0 - текущий день. 1 - завтра и т. д  
Возвращает строку

write\_schedule(login,pass,schedule,day)

schedule – двумерный массив 3\*n, n от 0 до 20. В первом столбце имя события, во втором время начала, в третьем время конца

В дальнейшем:   
get\_elective\_properties(elective\_id,day)  
get\_elective\_name(elective\_id,day)