



## **VII Поволжская открытая олимпиада по информационным технологиям «Волга ИТ – 2014»**

### **Номинация «Интернет-программирование (php)»**

#### **Задание финального этапа**

#### **Задание «Почта»**

##### **Исходные данные**

Участникам олимпиады предлагается организовать приложение отправки и контроля почты. Приложение состоит из одной страницы с блоками статистикой почтовых отправок и блока формы отправки нового отправления. Отправка ведется при помощи сервиса [mandrillapp.com](http://mandrillapp.com), который умеет как отсылать сообщения, так и отдавать статистику по ранее отправленным. Регистрация на [mandrillapp](http://mandrillapp.com) бесплатная, с ограничением в 12000 сообщений в месяц. Разработчик должен завести в [mandrillapp](http://mandrillapp.com) себе отдельный бесплатный аккаунт и использовать его в своем приложении.

##### **Блок статистики почтовых отправок**

Состоит из таблички с ранее отправленными почтовыми сообщениями, где для каждого сообщения показывается:

- дата отправки
- from (email)
- to (email)
- subject

- количество открытых сообщений
- количество кликов на сообщение

Сообщение показывается в этом блоке сразу после отправления.

### **Блок отправки сообщения**

Состоит из полей:

- From email
- To email
- Subject
- Body

+ кнопка отправки

Базовая задача (обязательная)

1. Реализовать оба блока

Максимальное количество баллов: 202

Дополнительные задачи

1. Для отрисовки страницы использовать bootstrap 3.x

Максимальное количество баллов: 40

2. Отправка сообщения и обновление списка сообщений делается через AJAX + обновление списка сообщений делается с частотой не менее 15 секунд.

Максимальное количество баллов: 100

3. В списке сообщений subject становится ссылкой, при нажатии на которую в роруп показывается текст сообщения.

Максимальное количество баллов: 80

4. Таблица с сообщениями получается сортировку по столбцам. Сортировка - чисто на javascript. Сортировка срабатывает при клике на заголовок столбца. При повторном клике на этот-же заголовок - происходит инверсия сортировки.

Максимальное количество баллов: 90

### **Задание “Редактирование CSV”**

Исходные данные

Участникам олимпиады предлагается организовать редактирование закачиваемого CSV-файла.

Редактор CSV-файла состоит из одной страницы, которая разделена на блоки:

#### **Блок загрузки файла**

Отвечает за загрузку CSV-файла, состоит из поля указания файла и кнопки “Загрузить”.

Загрузка происходит без проверки на тип файла - предполагается, что пользователь сам ответственен за загружаемый файл.

Разделитель строк в CSV-файле - запятая.

#### **Блок редактирования CSV-файла**

Показывается, только если файл загружен.

Состоит из таблички с данным из CSV-файла. Каждая ячейка - input поля.

Меня можно сразу все поля, все строчки. Сохранения - массово всего сразу.

### **Блок выгрузки отредактированного файла**

Показывается, только если файл загружен.

Состоит из кнопки “Выгрузить”, которая инициирует отдачу файла и из кнопки “Стереть файл”, которая забывает о редактируемом файле.

Базовая задача (обязательная)

1. Реализовать все 3 страницы
2. Обновление страницы сделать через AJAX

Максимальное количество баллов: 212

Дополнительные задачи

1. Для отрисовки страницы, используя bootstrap 3.x

Максимальное количество баллов: 40

2. Сделать отправку данных в блоке редактирования файла через AJAX.

Максимальное количество баллов: 85

3. Добавить возможность при загрузке файла выбрать его кодировку и разделитель столбцов. Выгружать файл надо с таким же разделителем и кодировкой.

Максимальное количество баллов: 85

3. Сделать возможность менять строки в таблице перетаскиванием. В результате - в выгружаемом файле эти строки тоже должны меняться местами.

Максимальное количество баллов: 90

---

## **Задание “Чат”**

Исходные данные

Участникам олимпиады предлагается организовать чат.

Чат состоит из страницы инициализации чата, страницы присоединения к чату и страницы самого чата.

### **Страница инициализации чата**

Состоит ровно поля “Имя пользователя” и кнопки “Начать чат”. После ее нажатия создается новый чат, с уникальным ключом и пользователь перебрасывается на страницу чата.

### **Страница присоединения к чату**

Если пользователь открыл страницу чата первый раз - у него спрашивается его имя, и если имя уникально - он попадает на страницу чата. Соответственно - страница состоит из поля “Имя пользователя” и кнопки “Присоединиться”.

### **Страница чата**

Состоит из надписи с именем пользователя, блока с историей чата и поля с новым сообщением. Когда вводится новое сообщение - он тут же появляется в истории у пользователя. У остальных пользователей оно появляется с задержкой не более 5 секунд и на своем месте. История показывается в виде списка сообщения, где каждое сообщение содержит имя пользователя, время сообщения и текст сообщения.

URL страницы можно выделить и послать другим пользователям. Они будут попадать в чат через страницы присоединения к чату.

Базовая задача (обязательная)

1. Реализовать все 3 страницы
2. Обновление страницы чата сделать через AJAX

Максимальное количество баллов: 232

Дополнительные задачи

1. Для отрисовки страниц использовать bootstrap 3.x

Максимальное количество баллов: 60

2. Для обновления страницы чата использовать не просто AJAX, а передачу данных через JSON. Причем - как для обновления истории, так и для отсылки нового сообщения.

Максимальное количество баллов: 100

3. Показывать пользователей, находящихся онлайн, на странице чата, с точностью в 30 секунд. То есть - если пользователь закрыл вкладку - то не позже, чем через 30 секунд он должен исчезнуть из списка активных пользователей у всех участников чата.

Максимальное количество баллов: 120