

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ**

**ОТЧЕТ
по лабораторной работе №3
по дисциплине «Web-технологии»
Тема: Модуль администрирования приложения «Аукцион картин»**

Студент гр.8303
Преподаватель

Рудько Д.Ю.
Беляев С.А.

Санкт-Петербург
2020

Цель работы.

Изучить возможности применения компилятора Babel, библиотеки jQuery, препроцессора LESS, инструмента выполнения повторяющихся задач GULP, регистрация разработанных модулей, формирование навыков построения структурированных web-приложений, освоение особенностей стандартных библиотек.

Задание.

Необходимо создать web-приложение, обеспечивающее администрирование аукциона картин: можно выбрать картины для участия в аукционе, определить начальные ставки, перечень участников и параметры аукциона. Основные требования следующие:

1. Перечень доступных картин с описаниями и ссылками на рисунки хранится в JSON-файле.
2. В качестве сервера используется Node.JS с модулем express.
3. Разработка ведётся с использованием стандарта не ниже ECMAScript2015.
4. Стили описываются с использованием LESS, при этом используются ключевые методы LESS (переменные, вложенные блоки, миксины, операторы и т. п.).
5. Клиентская часть разрабатывается с использованием jQuery (работа с DOM, AJAX-запросы).
6. Предусмотрена HTML-страница для перечня картин и карточка отдельной картины (название, автор, описание, изображение, начальная цена, минимальный и максимальный шаги аукциона). Предусмотрена возможность редактировать текстовые и числовые параметры, а также включить или исключить картину из участия в предстоящих торгах, загрузить рисунок картины.
7. Предусмотрена HTML-страница для списка потенциальных участников аукциона. Есть возможность добавлять или удалять участников, изменять запас денежных средств.
8. Предусмотрена HTML-страница для настроек аукциона (настройка даты и времени начала аукциона, настройка таймута продажи картины, настройка интервала времени отсчета до окончания торга по картине, паузы на изучение информации по картине для начала торга по ней).
9. Взаимодействие браузера с сервером осуществляется по протоколу HTTPS.
10. Сборка клиентской части (преобразования less, pug, babel, минификация) осуществляется с использованием GULP.
11. Регистрация и удаление разработанных модулей в npm.
12. Сохранение сформированных настроек в JSON-файл.

Ход выполнения.

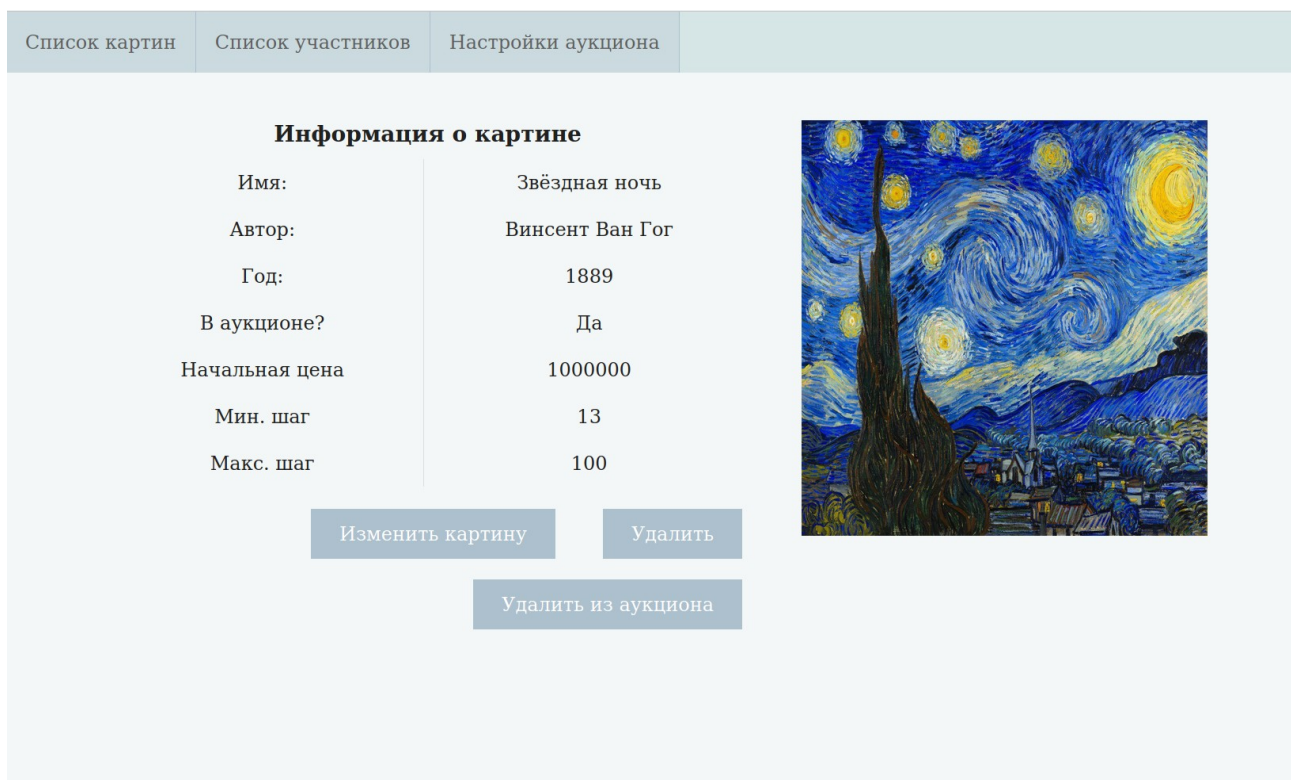
Создан web-сервер на основе express.

Написан шаблон для основной страницы, реализован ввод из json файла.

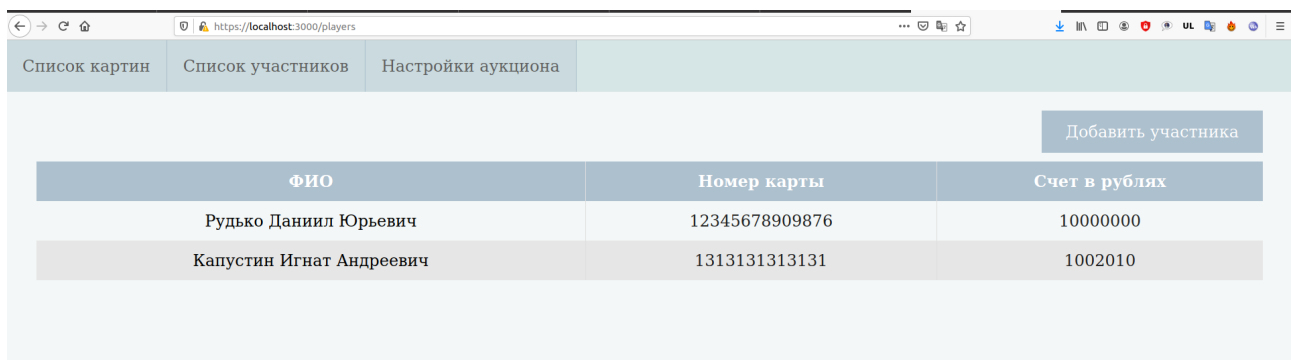
Разработан дизайн и стили главного окна и написаны методы для обработки всех необходимых кнопок главного окна.



реализована страница отдельной картины и написаны методы для обработки всех необходимых кнопок



реализована страница со списком участников и написаны методы для обработки всех необходимых кнопок



реализованы модальные окна для добавления участников и картин и написаны методы для обработки всех необходимых кнопок

Добавить участника

ФИО:

Номер карты:

Счет:

Отмена

Сохранить

Настройки аукциона

Добавить картину

Название:

Автор:

Год создания:

Ссылка на картинку:

Отмена

Сохранить

	Цена	Мин.
Винсе	000	13
Иван А		-
Иль	00	10

реализована страница с настройкой аукциона и модальное окно для изменения настроек и написаны методы для обработки всех необходимых кнопок

Список картин	Список участников	Настройки аукциона
---------------	-------------------	--------------------

Настройки аукциона

Дата начала:	2020-10-23
Время начала:	11:11
Таймаут продажи картины (мин.):	15
Интервал времени отсчета (мин.):	15
Пауза (мин.):	15
Картина:	Звёздная ночь

Изменить

Список картин	Список участников	Настройки аукциона
---------------	-------------------	--------------------

Изменить параметры аукциона

Дата начала:	23 . 10 . 2020
Время начала:	11 : 11
Таймаут продажи картины (мин.):	15
Интервал времени отсчета (мин.):	15
Пауза (мин.):	15
Название картины	Звёздная ночь

Сохранить

Сборка клиентской части осуществлена с использованием GULP.

Выводы.

В ходе выполнения лабораторной работы были изучены возможности применения компилятора Babel, библиотеки jQuery, препроцессора LESS и инструмента выполнения повторяющихся задач GULP. Получены навыки регистрации разработанных модулей и построения структурированных web-приложений. Освоены особенности стандартных библиотек.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД

www

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

var app = require('./app');
var debug = require('debug')('untitled:server');
var https = require('https');
var fs = require('fs');

/**
 * Get port from environment and store in Express.
 */

var hskey = fs.readFileSync('example.key');
var hscert = fs.readFileSync('example.csr')

var options = {
  key: hskey,
  cert: hscert
};

var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

/**
 * Create HTTP server.
 */

var server = https.createServer(options, app);

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  var port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}

/**
 * Event listener for HTTP server "error" event.
 */

function onError(error) {
  if (error.syscall !== 'listen') {
    throw error;
  }
}
```

```

var bind = typeof port === 'string'
  ? 'Pipe ' + port
  : 'Port ' + port;

// handle specific listen errors with friendly messages
switch (error.code) {
  case 'EACCES':
    console.error(bind + ' requires elevated privileges');
    process.exit(1);
    break;
  case 'EADDRINUSE':
    console.error(bind + ' is already in use');
    process.exit(1);
    break;
  default:
    throw error;
}
}

/**
 * Event listener for HTTP server "listening" event.
 */

function onListening() {
  var addr = server.address();
  var bind = typeof addr === 'string'
    ? 'pipe ' + addr
    : 'port ' + addr.port;
  debug('Listening on ' + bind);
}

```

app.js

```

var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/route');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'pug');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use('/jquery', express.static(__dirname + '/node_modules/jquery/dist/'));
app.use('/', indexRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

```

```
module.exports = app;
```

gulpfile.js

```
const gulp = require('gulp');
const less = require('gulp-less');
const babel = require('gulp-babel');
const cleanJS = require('gulp-uglify');
const cleanCSS = require('gulp-clean-css');
const pug = require('gulp-pug');
const del = require('del');

gulp.task('cleanJs', function(){
  return del('./public/GScripts/*.js');
});

gulp.task('makeJs', function(){
  return gulp.src('./public/javascripts/*.js')
    .pipe(babel())
    .pipe(gulp.dest('./public/GScripts/'));
});

gulp.task('cleanCss', function(){
  return del('./public/stylesheets/*.css');
});

gulp.task('makeCss', function(){
  return gulp.src('./public/stylesheets/*.less')
    .pipe(less())
    .pipe(cleanCSS())
    .pipe(gulp.dest('./public/stylesheets/'));
});

gulp.task("default", gulp.parallel((gulp.series('cleanJs', 'makeJs')),
  (gulp.series('cleanCss', 'makeCss'))));
```

route.js

```
var express = require('express');
var router = express.Router();
var fs = require('fs');
var path = require('path');

var arts = require('../public/json/arts.json');
var players = require('../public/json/players.json');
var auction = require('../public/json/auction.json');

/* GET домашняя стрвница (список картин). */
router.get('/', function(req, res, next) {
  res.render('paintings', { pictures: arts.arr});
});

/* GET переход к карточке картины. */
router.get('/painting/:id', function (req, res){
  const id = req.params.id *1;
  var picture = arts.arr.find(p => p.id === id);
  if (picture) {
    res.render('painting', { title: 'Картина', picture: picture});
  }
});

/* GET список игроков ( участников аукциона). */
router.get('/players', function (req, res){
  res.render('players', {participants: players.arr});
});

/* GET переход к карточке игрока (участника аукциона). */
router.get('/player/:id', function (req, res) {
  const id = req.params.id;
  var player = players.arr.find(p => p.id == id);
  if (player) {
    res.render('player', { title: 'Участник', participant: player,
  pictures:arts.arr});
  }
});
```



```

    }
  });

  /* GET страница настроек аукциона. */
  router.get('/ASettings', (req, res, next) => {
    res.render('ASettings', {title: 'Настройки аукциона', pictures: arts.arr,
    auction: auction.myAuc});
  });

  /* добавление картины. */
  router.get('/addPic', function (req, res){
    console.log("IN ROUTERS");
    let id = arts.arr.length+1;
    let newPic = {
      "id": id,
      "name": req.query.name,
      "author": req.query.author,
      "date": req.query.date,
      "beginning_price": "-",
      "pt": req.query.pt,
      "inAuction": false,
      "min_step": "-",
      "max_step": "-"
    }
    arts.arr.push(newPic);

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'arts.json'), JSON.stringify(arts), function(err) {
      if(err) throw err;
    })

    return res.send(newPic);

  });

  /* добавление участника. */
  router.get('/addPl', function (req, res){
    let id = players.arr.length;
    let newPl = {
      "id": id,
      "name": req.query.name,
      "card": req.query.card,
      "money": req.query.money,
      "auction": ""
    }
    players.arr.push(newPl);

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'players.json'), JSON.stringify(players), function(err) {
      if(err) throw err;
    })

    return res.send(newPl);

  });

  /* изменение настроек аукциона. */
  router.get('/changeSet', function (req, res){
    var auc = req.query;
    auc.id="auction";
    auction.myAuc = auc;

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'auction.json'), JSON.stringify(auction), function(err) {
      if(err) throw err;
    })

    return res.send(auction.myAuc);
  });

```

```

});

/* изменения картины. */
router.get('/changePic', function (req, res){
  console.log("IN CHANGE");
  var pic = req.query;
  let id = req.query.id *1;
  var index = -1;
  for(let i = 0; i < arts.arr.length; i++) {
    if (arts.arr[i].id === id) {
      index = i;
    }
  }
  if (arts.arr[index].inAuction === true) {
    arts.arr[index].name = req.query.name;
    arts.arr[index].author = req.query.author;
    arts.arr[index].date = req.query.date;
    arts.arr[index].beginning_price = req.query.beginning_price;
    arts.arr[index].min_step = req.query.min_step;
    arts.arr[index].max_step = req.query.max_step;
    pic.inAuction = true;
  } else {
    arts.arr[index].name = req.query.name;
    arts.arr[index].author = req.query.author;
    arts.arr[index].year = req.query.year;
    pic.inAuction = false;
  }

  //console.log(arts.arr[index]);

  pic.functionName = 'changePic';

  fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'arts.json'), JSON.stringify(arts), function(err) {
    if(err) throw err;
  })

  //console.log(pic);

  return res.send(pic);

});

/* удаления картины со ставок. */
router.get('/removeFromAuction', function (req, res){
  let id = req.query.id *1;
  var pic = req.query;
  var index = -1;
  for(let i = 0; i < arts.arr.length; i++) {
    if (arts.arr[i].id === id) {
      index = i;
    }
  }
  arts.arr[index].inAuction = false;

  pic.functionName = 'removeFromAuction';

  fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'arts.json'), JSON.stringify(arts), function(err) {
    if(err) throw err;
  })

  return res.send(pic);

});

/* удаления картины. */
router.get('/delPic', function (req, res){
  var pic = req.query;
  let id = pic.id *1;

```

```

var index = -1;
for(let i = 0; i < arts.arr.length; i++) {
    if (arts.arr[i].id === id) {
        index = i;
    }
}
if (index !== -1) {
    arts.arr.splice(index, 1);
}

fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'arts.json'), JSON.stringify(arts), function(err) {
    if(err) throw err;
})

pic.functionName = 'delPic';

return res.send(pic);
})

/* добавление картины на ставки. */
router.get('/addInAuction', function (req, res){
    var pic = req.query;
    var index = -1;
    for(let i = 0; i < arts.arr.length; i++) {
        if (arts.arr[i].id === (pic.id*1)) {
            index = i;
            break;
        }
    }
    arts.arr[index].inAuction = true;
    arts.arr[index].beginning_price = pic.beginning_price;
    arts.arr[index].min_step = pic.min_step;
    arts.arr[index].max_step = pic.max_step;
    pic = arts.arr[index];

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'arts.json'), JSON.stringify(arts), function(err) {
        if(err) throw err;
    })

    pic.functionName = 'addInAuction';

    console.log(pic);

    return res.send(pic);
});

/* удаления игрока. */
router.get('/delPl', function (req, res){
    var pl = req.query;
    let id = pl.id *1;
    var index = -1;
    for(let i = 0; i < players.arr.length; i++) {
        if (players.arr[i].id === id) {
            index = i;
        }
    }
    if (index !== -1) {
        players.arr.splice(index, 1);
    }

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'players.json'), JSON.stringify(players), function(err) {
        if(err) throw err;
    })

    return res.send(pl);
}

```

```

});

/* добавление картины в желание игрока. */
router.get('/addPicToPl', function (req, res){
    var pl = req.query;
    let id = pl.id*1;
    var index = -1;
    for(let i = 0; i < players.arr.length; i++) {
        if (players.arr[i].id === id) {
            index = i;
            break;
        }
    }
    if(players.arr[index].auction === "")
        players.arr[index].auction = []
    players.arr[index].auction.push({picture: pl.picture, maxPrice: pl.maxPrice});

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'players.json'), JSON.stringify(players), function(err){
        if(err) throw err;
    })

    return res.send(pl);
});

/* изменение денежного счета игрока. */
router.get('/changeMoney', function (req, res){
    var pl = req.query;
    let id = pl.id *1;;
    let index = -1;
    for(let i = 0; i < players.arr.length; i++) {
        if (players.arr[i].id === id) {
            index = i;
            break;
        }
    }
    players.arr[index].money = pl.money;

    fs.writeFile(path.join(__dirname, '..', 'public', 'json', 'players.json'), JSON.stringify(players), function(err){
        if(err) throw err;
    })

    return res.send(pl);
})
module.exports = router;

```

art.js

```

var id = document.location.href.split('/painting/')[1].split('#')[0];

$(document).ready(()=> {
    // изменить инлрмацию о картине
    $('#__save').click(()=>{

        if($('#changeMin_').val()*1 < 0){
            alert("Допускаются только положительные значения\nЗначение заменено на положительное");
            let dop = $('#changeMin_').val()*-1;
            $('#changeMin_').val(dop);
        }

        if($('#changeMin_').val()*1 > $('#changeMax_').val()*1){
            alert("Начение минимального шага должно быть меньше чем значение максимального\nЗначение заменены");
            let dopMin = $('#changeMin_').val();
            let dopMax = $('#changeMax_').val();
            $('#changeMin_').val(dopMax);

```

```

        $('#changeMax_').val(dopMin);
    }

    var formData = {
        id: id,
        name: $('#changeName').val(),
        author: $('#changeAuthor').val(),
        date: $('#changeYear').val(),
        description: $('#changeDes').val(),
        inAuction: null,
        beginning_price: $('#changePrice_').val(),
        min_step: $('#changeMin_').val(),
        max_step: $('#changeMax_').val()
    }

    sendAjax('/changePic', formData);

});
// удалить картину из аукциона
$('#removePicButton').click(()=>{
    var data = {id: id}
    sendAjax('/removeFromAuction', data);
});
// удалить картину
$('#modalDelete').click(()=>{
    var data = {id: id};
    sendAjax('/delPic', data);
});
// добавить картину в аукцион
$('#save').click(()=>{

    if($('#changeMin').val()*1 < 0){
        alert("Допускаются только положительные значения\nЗначение заменено
на положительное");
        let dop = $('#changeMin').val()*-1;
        $('#changeMin').val(dop);
    }

    if($('#changeMin').val()*1 > $('#changeMax').val()*1){
        alert("Начение минимального шага должно быть меньше чем значение
максимального\nЗначение заменены");
        let dopMin = $('#changeMin').val();
        let dopMax = $('#changeMax').val();
        $('#changeMin').val(dopMax);
        $('#changeMax').val(dopMin);
    }

    var formData1 = {
        id: id,
        name: $('#changeName').val(),
        author: $('#changeAuthor').val(),
        date: $('#changeYear').val(),
        description: $('#changeDes').val(),
        inAuction: null,
        beginning_price: $('#changePrice').val(),
        min_step: $('#changeMin').val(),
        max_step: $('#changeMax').val()
    }

    sendAjax('/addInAuction', formData1);
});
});

function sendAjax(url, data){
    $.ajax({
        url: url,
        type: "GET",
        data: data,
        dataType: "json",
        contentType: "application/json",
        async: false,
        error: function (e){

```

```

        console.log(data);
        console.log("ERROR: ", e.status);
    },
    success: foo
});

function foo(data){
    //alert(data.functionName);
    if(data.functionName === 'changePic'){
        $('#td1').text($('#changeName').val());
        $('#td2').text($('#changeAuthor').val());
        $('#td3').text($('#changeYear').val());
        if(data.inAuction === true) {
            $('#td5').text('Да');
            $('#td6').text($('#changePrice_').val());
            $('#td7').text($('#changeMin_').val());
            $('#td8').text($('#changeMax_').val());
        }
    }
    if(data.functionName === 'delPic'){
        setTimeout(function() {document.location.href='/';}, 400);
    }
    if(data.functionName === 'removeFromAuction'){
        $('#tr5').remove();
        $('#tr6').remove();
        $('#tr7').remove();
        $('#tr8').remove();
        $('#addAuction').text('Поставить в аукцион');
        $('#addAuction').attr({form:'addPicToAuc', href:'#modalAddPic'});
        setTimeout(function() {document.location.href='/painting/${id}`;}, 200);
    }
    if(data.functionName === 'addInAuction'){
        $('#addAuction').text('Удалить из аукциона');
        $('#addAuction').attr({form:'removePicFromAuc',
href:'#modalRemovePic'});
        $('#tab tr:last').after('<tr id="tr5">' + '<td>' + 'В аукционе?' + '<td id="td5">' + 'Да' + '</tr>' + '<tr id="tr6">' +
            '<td>' + 'Начальная цена' + '<td id="td6">' + $
            ('#changePrice').val() + '</tr>' + '<tr id="tr7">' +
            '<td>' + 'Мин. шаг' + '<td id="td7">' + ($('#changeMin').val() +
            '</tr>' + '<tr id="tr8">' +
            '<td>' + 'Макс. шаг' + '<td id="td8">' + ($('#changeMax').val() +
            '</tr>');
        setTimeout(function() {document.location.href='/painting/${id}`;}, 200);
    }
}

```

arts.js

```

$(document).ready(()=>{
    $('#save').click(()=>{
        var formData = {
            name: $('#newPicName').val(),
            author: $('#newPicAuthor').val(),
            date: $('#newPicDate').val(),
            description: $('#newPicDescription').val(),
            pt: $('#newPicImage').val(),
            inAuction: false,
            beginning_price: "-",
            min_step: "-",
            max_step: "-"
        }
        sendAjax('/addPic', formData);
    });
});

function sendAjax(url, formData) {
    $.ajax({
        url: url,
        type: "GET",

```

```

        data: formData,
        dataType: "json",
        contentType: "application/json",
        async: false,
        error: function (e){
            console.log(formData);
            console.log("ERROR: ", e.status);
        },
        success: foo
    });
}

function foo(data) {
    //alert(JSON.stringify(data));

    $('#tbody tr:last').after('<tr>'+<td>' + '<a href=/painting/' +data.id +
    '>'+$('#newPicName').val() + '<td>'+ $('#newPicAuthor').val() +
        '<td>' + $('#newPicDate').val() + '<td>' + "Her" + '<td>' + "-" + '<td>' +
    "-" + '<td>' + "-" + '</tr>');
}
$('#_save').click();

```

auctionSettings.js

```

$(document).ready(()=> {
    $('#save').click(()=> {
        sendAjax('/changeSet')
    })
});

function sendAjax(url){

    var formData = {
        dateBegin: $('#dateBegin').val(),
        timeBegin: $('#timeBegin').val(),
        timeout: $('#timeout').val(),
        interval: $('#interval').val(),
        pause: $('#pause').val(),
        picName: $('#addNewPic').val(),
    };

    $.ajax({
        url:url,
        type: "GET",
        data: formData,
        dataType: "json",
        contentType: "application/json",
        async: false,
        error: function (e){
            console.log(formData);
            console.log("ERROR: ", e.status);
        },
        success: foo
    });
}

function foo(data){
    $('#td1').text($('#dateBegin').val());
    $('#td2').text($('#timeBegin').val());
    $('#td3').text($('#timeout').val());
    $('#td4').text($('#interval').val());
    $('#td5').text($('#pause').val());
    $('#td6').text($('#addNewPic').val());
}

```

player.js

```

var id = document.location.href.split('/player/')[1].split('#')[0];

$(document).ready(()=> {
    // Удалить игрока
    $('#modalDelete').click(() => {
        var data = { id: id};
    });
});

```

```

        sendAjax('/delPl', data);
        setTimeout(function() {document.location.href='/players';}, 400);
    });
    // Добавить картину вжелания
    $('#__save').click(()=>{
        var data = {id: id, picture: $('#addNewPic').val(), maxPrice: $('#addPartMoney').val()}
        sendAjax('/addPicToPl', data);
        $("#tbody tr:last").after('<tr>'+ '<td>' + $('#addNewPic').val() +
'<td>' + $('#addPartMoney').val() + '</tr>');
        setTimeout(function() {document.location.href='/player/'+id;});
    })
    //изменить счет игрока
    $('#__save').click(()=>{
        var data = {id:id, money: $('#changePartMoney').val()}
        sendAjax('/changeMoney', data);
        $('#money').text($('#changePartMoney').val())
    })
});

function sendAjax(url, data){
    $.ajax({
        url: url,
        type: "GET",
        data: data,
        dataType: "json",
        contentType: "application/json",
        async: false,
        error: function (e){
            console.log(data);
            console.log("ERROR: ", e.status);
        },
        success: foo
    });
}

function foo(){
    console.log("Success");
}

```

players.js

```

$(document).ready(()=>{
    $('#save').click(()=>{
        sendAjax('/addPl');
    });
});

function sendAjax(url) {

    var formData = {
        name: $('#newParticipantName').val(),
        card: $('#newParticipantCard').val(),
        money: $('#newParticipantMoney').val(),
        auction: ""
    }

    $.ajax({
        url: url,
        type: "GET",
        data: formData,
        dataType: "json",
        contentType: "application/json",
        async: false,
        error: function (e){
            console.log(formData);
            console.log("ERROR: ", e.status);
        },
        success: foo
    });
}

function foo(data) {

```



```

    $('#tbody tr:last').after('<tr>'+'<td>' + '<a href=/player/' +data.id + '>'
+ $('#newParticipantName').val() + '<td>' + $('#newParticipantCard').val() +
    '<td>' + $('#newParticipantMoney').val() + '</tr>');
}
$('#_save').click();

```

style.less

```

@backgroundColor: #F3F7F7;
@mainTextColor: #222222;
@backgroundTableColor: #ADC0CE;
@TitleTextColor: #ffffff;
@firstChildBackgroundTableColor: #F3F7F7;
@secondChildBackgroundTableColor: #e6e6e6;
@hoverBackgroundTableColor: #ADC0CE;
@menuBackgroundColor: #D6E5E5;
@menuTextColor: #606060;
@menuHoverBackgroundColor: #A4A4EA;
@menuActiveBackgroundColor: #ADC0FF;
@Georgia: Georgia, serif;

.fooFloatPadding(@x:right, @y:10px) {
    float: @x;
    padding: @y;
}

html, body {
    height: 100%;
}

* {
    margin: 0; /*отступ*/
}

body {
    background: @backgroundColor;
    color: @mainTextColor;
    font-family: @Georgia;
    display: flex;
    flex-direction: column;
}

.content, .header, .footer {
    box-sizing: border-box;
    width: 100%;
}

.footer {
    left: 0; bottom: 0;
    padding: 20px;
    font-family: @Georgia;
    p {
        margin: 10px 0;
        font-family: @Georgia;
        font-size: 15px;
    }
    a {
        color: @menuTextColor;
    }
}

.header, .footer {
    background: @menuBackgroundColor;
}

.content {
    flex-grow: 1;
    padding: 20px;
    p {
        margin: 10px 0;
        font-size: 16px;
        line-height: 1.8;
    }
}

```

```

    }
}

.content table {
    width: 100%;
    text-align: center;
    border-radius: 6px;
    border-collapse: separate;
    border-spacing: 0px;
    padding: 10px;
    thead tr {
        color: @TitleTextColor;
        font-weight: bold;
        background: @backgroundTableColor;
        padding: 10px;
    }
    tr {
        td {
            border-right: 1px solid #dfdfdf;
            padding: 10px;
            a {
                text-decoration: none;
                color: #000000;
            }
        }
        td:last-child {
            border-right: 0px;
        }
        th {
            border-right: 1px solid #dfdfdf;
            padding: 10px;
        }
        th:last-child {
            border-right: 0px;
        }
    }
    tbody {
        tr:nth-child(1n) {
            background: @backgroundColor;
        }
        tr:hover {
            background: @hoverBackgroundTableColor;
            transition-duration: 0.6s;
        }
    }
}

.content .m_table tbody tr:nth-child(2n) {
    background: @secondChildBackgroundTableColor;
}

.content .m_table tbody tr:hover {
    background: @hoverBackgroundTableColor;
    transition-duration: 0.6s;
}

.menu {
    nav {
        display: block;
        width: 660px;
        margin: 0 auto 30px;
    }
    ul {
        list-style: none;
        margin: 0 auto;
        background: @menuBackgroundColor;
        overflow: hidden;
        padding: 0;
    }
    li {
        float: left;

```

```

}
a {
  text-decoration: none;
  display: block;
  font-family: @Georgia;
  transition: .5s linear;
  padding: 1em;
  border-right: 1px solid @backgroundTableColor;
  background: rgba(173, 192, 206, .3);
  color: @menuTextColor;
}
a:hover {background: @backgroundTableColor;}
a:active {background: @backgroundTableColor;}
}

.modal {
  background: @backgroundColor;
  left: 50%;
  margin: -250px 0 0 -23%;
  position: absolute;
  top: -50%;
  width: 40%;
  border-radius: 5px;
  box-shadow: 0 3px 7px rgba(0,0,0,.25);
  -moz-box-shadow: 0 3px 7px rgba(0,0,0,.25);
  -webkit-box-shadow: 0 3px 7px rgba(0,0,0,.25);
  box-sizing: border-box;
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
}

/* Активация модального окна в состоянии :target */
.modal:target {
  top: 50%;
  transition: all 0.4s ease-in-out;
  -moz-transition: all 0.4s ease-in-out;
  -webkit-transition: all 0.4s ease-in-out;
}

.modal header, .modal footer {
  background: #D6E5E5;
  border-bottom: 1px solid #e7e7e7;
  border-radius: 5px 5px 0 0;
  -moz-border-radius: 5px 5px 0 0;
  -webkit-border-radius: 5px 5px 0 0;
}

.modal footer {
  height: 40px;
  border: none;
  border-top: 1px solid #e7e7e7;
  border-radius: 0 0 5px 5px;
  -moz-border-radius: 0 0 5px 5px;
  -webkit-border-radius: 0 0 5px 5px;
}

.modal section, .modal header, .modal footer {
  padding: 15px;
  z-index: 200;
}

input {
  border-radius: 3px;
  outline: none;
  max-width: 180px;
}

.buttons .add, .filter {
  .fooFloatPadding();
}

.buttons .save, .cancel {
  .fooFloatPadding(left, 10px);
}

```

```

.buttons .change, .give, .take, .deleted {
    .fooFloatPadding(right,20px)
}

a.buttons {
    color: #fff; /* цвет текста */
    text-decoration: none; /* убирать подчёркивание у ссылок */
    user-select: none; /* убирать выделение текста */
    background: #ADC0CE; /* фон кнопки */
    padding: .7em 1.5em; /* отступ от текста */
    outline: none; /* убирать контур в Mozilla */
}
a.buttons:hover { background: @menuHoverBackgroundColor; } /* при наведении
курсора мышки */
a.buttons:active { background: @menuActiveBackgroundColor; } /* при нажатии */

img.indBook {
    -webkit-transition-duration: 0.5s; /*Webkit: длительность анимации*/
    -moz-transition-duration: 0.5s; /*Mozilla: длительность анимации*/
    -o-transition-duration: 0.5s; /*Opera: длительность анимации*/
    height: 80%;
    width: 60%;
}

.right-panel-wrapper {
    width: 800px;
}

.mcontent-wrapper {
    width: calc(100% - 400px);
    padding: 0 30px;
}

.page-wrapper {
    display: flex;
    justify-content: space-between;
}

.content .cardBook {
    width: 500px;
    padding: 20px;
    margin: auto;
}

```

Asettings.pug

extends layout

```

block content
    include ./header
    div(class='content')
        aside(class="modal" id="auction")
            header
                h3 Изменить параметры аукциона
            section
                form(id="delete_par_", action="#", method="post")
                    table
                        tr
                            td Дата начала:
                            td
                                input(type="date" id="dateBegin" value =
auction["dateBegin"])
                        tr
                            td Время начала:
                            td
                                input(type="time" id="timeBegin" value =
auction["timeBegin"])
                        tr
                            td Таймаут продажи картины (мин.):
                            td

```

```

                                input (type="number" id="timeout" value =
auction["timeout"])
                                tr
                                td Интервал времени отсчета (мин.):
                                td
                                input (type="number" id="interval" value =
auction["interval"])
                                tr
                                td Пауза (мин.):
                                td
                                input (type="number" id="pause" value =
auction["pause"])
                                tr
                                td Название картины
                                td
                                select#addNewPic
                                for pic in pictures
                                if (pic["inAuction"] == true)
                                option #{pic["name"]}

                                footer
                                div (class="buttons" align="center")
                                a (class="buttons" id="save" href="#modalSave"
form="delete_par") Сохранить
                                div (class='cardBook')
                                h4 (align="center") Настройки аукциона
                                table
                                tr
                                td Дата начала:
                                td#td1 #{auction["dateBegin"]}

                                tr
                                td Время начала:
                                td#td2 #{auction["timeBegin"]}

                                tr
                                td Таймаут продажи картины (мин.):
                                td#td3 #{auction["timeout"]}

                                tr
                                td Интервал времени отсчета (мин.):
                                td#td4 #{auction["interval"]}

                                tr
                                td Пауза (мин.):
                                td#td5 #{auction["pause"]}

                                tr
                                td Картина:
                                td#td6 #{auction["picName"]}
                                div (class="buttons")
                                div (class="add")
                                a (href='#auction', class="buttons", id="_add"
form="delete_par_") Изменить
                                script
                                include ../public/GScripts/auctionSettings.js

```

error.pug

```
extends layout
```

```

block content
  h1= message
  h2= error.status
  pre #{error.stack}

```

header.pug

```

div (class="header")
  nav (class="menu")
    ul
      li: a (href='/') Список картин
      li: a (href='/players') Список участников
      li: a (href='/ASettings') Настройки аукциона

```

index.pug

```
extends layout

block content
  h1= title
  p Welcome to #{title}
```

layout.pug

```
doctype html
html
  head
    title= title
    script
      include ../node_modules/jquery/dist/jquery.js
    link(rel='stylesheet', href='/stylesheets/style.css')
  body
    block content
```

painting.pug

```
extends layout

block content
  include ./header
  div(class='content')
    div(class="content")
      aside(class="modal" id="modalAddPic")
        header
          h3 Добавить в аукцион
        section
          form(id="addPicToAuc", action="#", method="GET")
            table
              tr
                td Начальная цена:
                td
                  input(type="text" id="changePrice")
              tr
                td Минимальный шаг:
                td
                  input(type="text" id="changeMin")
              tr
                td Максимальный шаг:
                td
                  input(type="text" id="changeMax")
            footer
              div(class="buttons")
                div(class="cancel")
                  a(href='#' class="buttons" id="cancel") Отмена
                div(class="save")
                  a(href='#' class="buttons" id="save")
          form="addPicToAuc") Сохранить
      aside(class="modal" id="modalRemovePic")
        header
          h3 Удаление картины из аукциона
        section
          form(id="removePicFromAuc", action="#", method="GET")
            p Вы уверены?
        footer
          div(class="buttons")
            div(class="cancel")
              a(href='#' class="buttons" id="cancelAuc") Нет
            div(class="save")
              a(href='#' class="buttons" id="removePicButton")
          form="modalRemovePic") Да
      aside(class="modal" id="modalDeletePic")
        header
          h3 Удаление картины
        section
          form(id="delete_pic", action="#", method="delete")
            p Вы уверены?
        footer
```

```


(class="buttons")


(class="cancel")
a(href="#" class="buttons" id="__cancel") Нет


(class="save")
a(href="#" class="buttons" id="modalDelete"
form="delete_pic") Да
aside(class="modal" id="modalChangePic")
header
h3 Изменить картину
section
form(id="change_pic", action="/changePic", method="GET")
table
tr
td Название:
td
(class="cancel")
a(href="#" class="buttons" id="__cancel") Отмена


(class="save")
a(href="#" class="buttons" id="__save"
form="change_pic") Сохранить


(class='page-wrapper')


(class='mcontent-wrapper')


(class="cardPic")
h3(align="center") Информация о картине
table#tab
tr
td Имя:
td#td1 #{picture["name"]}
tr
td Автор:
td#td2 #{picture["author"]}
tr
td Год:
td#td3 #{picture["date"]}
tr
td if (picture["inAuction"] === true)
tr#tr5
td В аукционе?
td#td5 Да
tr#tr6
td Начальная цена


```

```
 #td6 #{picture["beginning_price"]} |
```

| Мин. шаг | #td7 #{picture["min_step"]} |
| Макс. шаг | #td8 #{picture["max_step"]} |

[Удалить](#modalDeletePic)

[Изменить картину](#modalChangePic)

[Поставить в аукцион](#modalAddPic)

[Удалить](#modalDeletePic)

[Изменить картину](#modalChangePic)

[Удалить из аукциона](#modalRemovePic)

paintings.pug

```

extends layout
block content
  include ../header
  div(class="content")
    aside(class="modal" id="modal")
      header
        h3 Добавить картину
      section
        form(id="new_pic")
          table
            tr
              td Название:
  |

            tr
              td Автор:
  |

            tr
              td Год создания:
  |

            tr
              td Ссылка на картинку:
  |

          footer
            div(class="buttons")
              div(class="cancel")
                a(href="/" class="buttons" id="cancel">Отмена
              div(class="save")
                a(href="/" class="buttons" id="save" form="new_pic">Сохранить
            div(class="buttons")
              div(class="add")
                a(href="#modal", class="buttons", id="_add">Добавить картину

```



```

div( style="clear: both")
table(class="m_table" id ="m_table")
  thead
    tr
      th(datatype="title_name") Название
      th(datatype="title_author") Автор
      th(datatype="title_money") Год
      th(datatype="title_in_auction") В аукционе?
      th(datatype="title_begin_price") Начальная цена
      th(datatype="title_min_step") Мин. шаг
      th(datatype="title_max_step") Макс. шаг
  tbody#tbody
    for picture in pictures
      tr
        td
          a(href="/painting/"+picture["id"])=picture["name"]
        td #{picture["author"]}
        td #{picture["date"]}
        if (picture["inAuction"] == false)
          td Нет
          td -
          td -
          td -
        else
          td Да
          td #{picture["beginning_price"]}
          td #{picture["min_step"]}
          td #{picture["max_step"]}

script
  include ../public/GScripts/arts.js

```

player.pug

extends layout

```

block content
  include ./header
  div(class='content')
    div(class="content")
      aside(class="modal" id="modalDeletePar")
        header
          h3 Удаление участника аукциона
        section
          form(id="delete_par", action="#", method="post")
            p Вы уверены?
        footer
          div(class="buttons")
            div(class="cancel")
              a(href='#' class="buttons" id="___cancel") Нет
            div(class="save")
              a(href='#' class="buttons" id="modalDelete")
form="delete_par") Да
      aside(class="modal" id="modalChangePar")
        header
          h3 Изменить счет
        section
          form(id="change_book", action="#", method="post")
            table
              tr
                td Новый счет:
                td
                  input(type="text" id="changePartMoney")
        footer
          div(class="buttons")
            div(class="cancel")
              a(href='#' class="buttons" id="___cancel") Отмена
            div(class="save")
              a(href='#' class="buttons" id="___save")
form="change_par") Сохранить
      aside(class="modal" id="modalAddPic")
        header
          h3 Добавить картину
        section

```

```

        form(id="add_pic", action="#", method="put")
        table
            tr
                td Название картины
                td
                    select#addNewPic
                    for pic in pictures
                        option #{pic["name"]}
            tr
                td Максимальная сумма:
                td
                    input(type="text" id="addPartMoney")
    footer
        div(class="buttons")
        div(class="cancel")
            a(href="#" class="buttons" id="____cancel") Отмена
        div(class="save")
            a(href="#" class="buttons" id="____save")
form="add_pic") Сохранить
    div(class="cardPar")
        h3(align="center") Информация об участнике аукциона
        br
        div(class="buttons")
            div(class="deleted")
                a(href='#modalDeletePar' class="buttons"
id="deleteButton" form="delete_par") Удалить
            div(class="change")
                a(href='#modalChangePar' class="buttons" id="change"
form="change_par") Изменить счет
            div(class="change")
                a(href='#modalAddPic' class="buttons" id="addPic"
form="add_par") Добавить картину в желания
        table
            tr
                td ФИО:
                td #{participant["name"]}
            tr
                td Номер карты:
                td #{participant["card"]}
            tr
                td Счет:
                td#money #{participant["money"]}
        br
        if (participant["auction"]!="")
            h4(align="center") Желаемые картины
            table#mtable
                thead
                    tr
                        th(datatype="title_name") Название картины
                        th(datatype="title_author") Максимальная сумма
                tbody#mbody
                    for pic in participant["auction"]
                        tr
                            for p in pictures
                                if(p["name"] == pic["picture"])
                                    td
                                        a(href="/painting/" +
p["id"])=p["name"]
                                        td #{pic["maxPrice"]}
        br
    script
        include ../public/GScripts/player.js

```

players.pug

extends layout

```

block content
    include header
    div(class="content")
        aside(class="modal" id="modal")
            header

```

```

        h3 Добавить участника
    section
        form(id="new_participant", action="#", method="POST")
            table
                tr
                    td ФИО:
                    td
                        input(type="text" id="newParticipantName")
                tr
                    td Номер карты:
                    td
                        input(id="newParticipantCard" type="text" )
                tr
                    td Счет:
                    td
                        input(id="newParticipantMoney" type="text")
            footer
                div(class="buttons")
                    div(class="cancel")
                        a(href='/players' class="buttons" id="cancel") Отмена
                    div(class="save")
                        a(href='/players' class="buttons" id="save"
form="new_book") Сохранить
                div(class="buttons")
                    div(class="add")
                        a(href='#modal',class="buttons",id="_add") Добавить участника
            div( style="clear: both")
            table(class="m_table" id ="m_table")
                thead
                    tr
                        th(datatype="title_name") ФИО
                        th(datatype="title_card") Номер карты
                        th(datatype="title_money") Счет в рублях
                tbody#tbody
                    for participant in participants
                        tr
                            td

a(href="/player/"+participant["id"])=participant["name"]
                            td #{participant["card"]}
                            td #{participant["money"]}
            script
                include ../public/GScripts/players.js

```