

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Житомирський державний технологічний університет

Кафедра програмного
забезпечення систем
Група ПІ-48

Лабораторна робота `hall of fame`

Виконали:

Бліндарук

Перевірила:

Данильченко А.О.

Житомир 2016

Створення додатку 'hall of fame'

Додаток створений для відображення студентів ФІКТу, які отримати середній бао 5.0 за результатами останньої сесії.

<http://halloffame-networkofmentors.rhcloud.com/>

Скріншоти програми

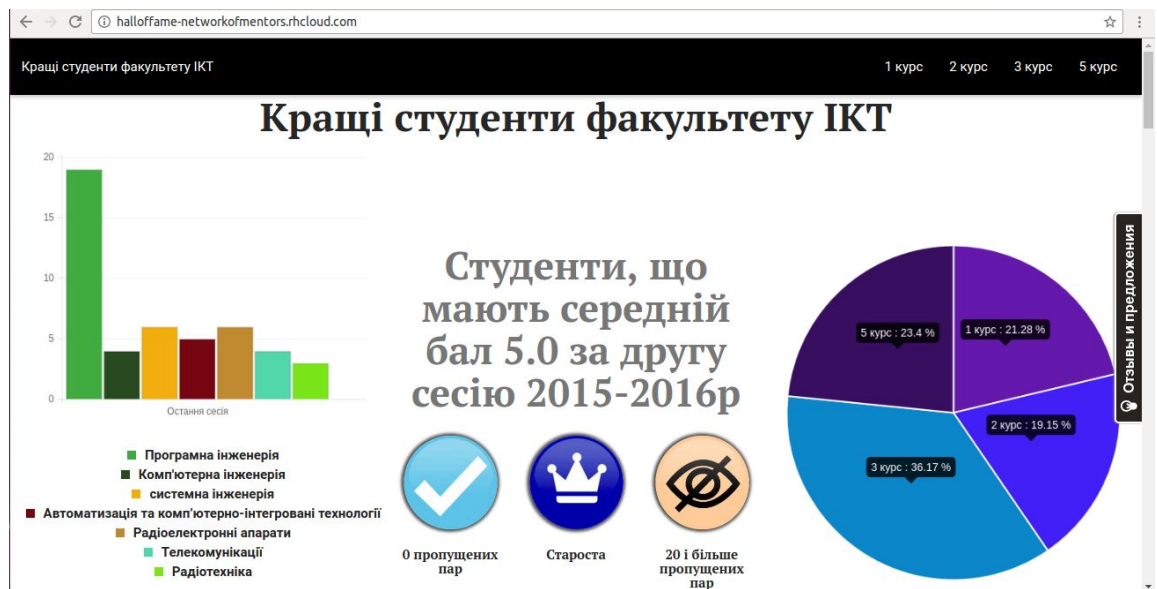


Рис 1 . Статистика по специальностям

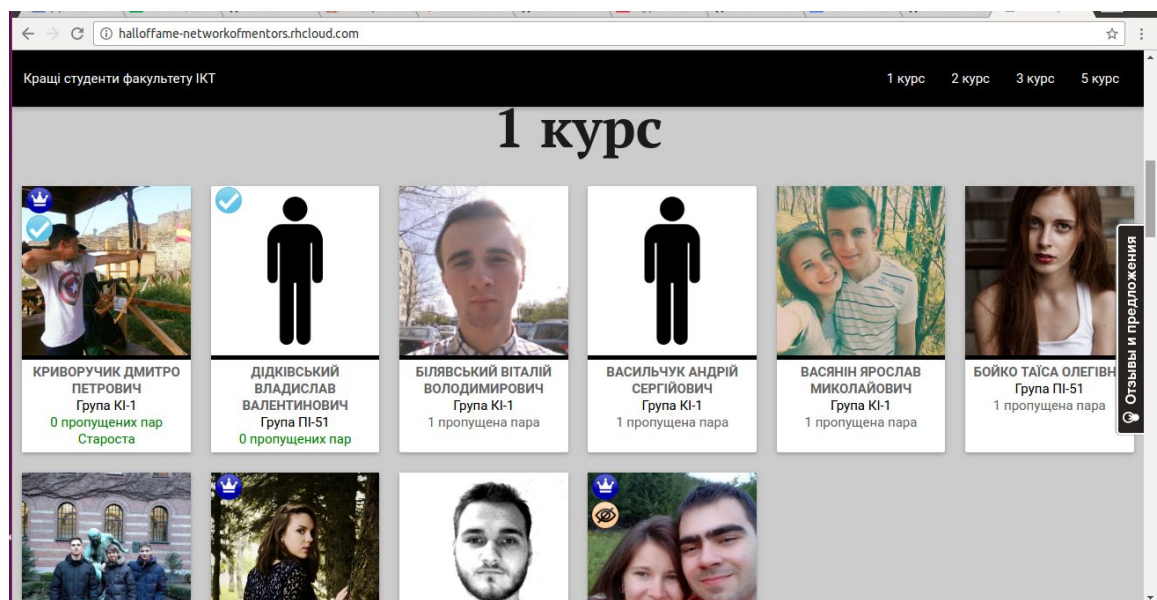


Рис 2. Приклад виводу найліпших студентів

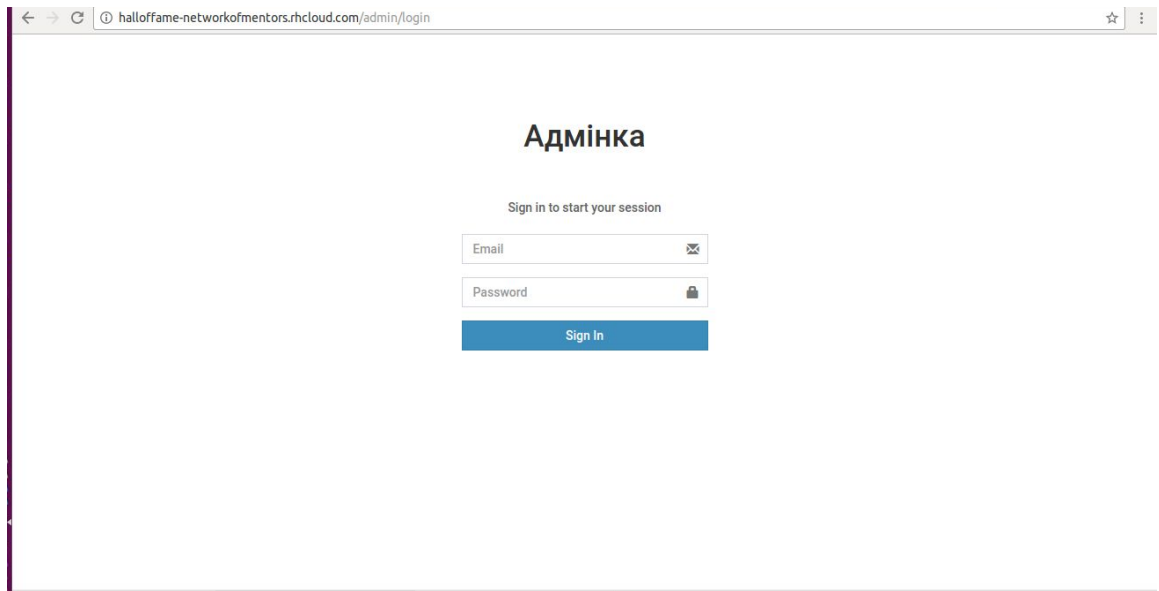


Рис 3. Логін в Адмінку

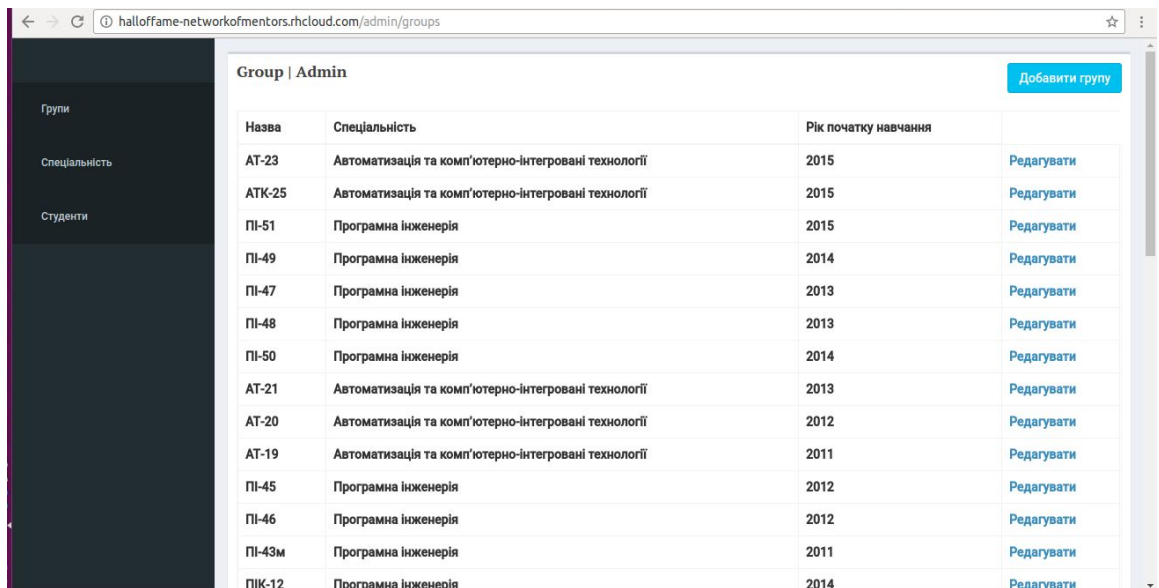


Рис 3. Адмінка

Код програми

server.js

```
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var session = require('express-session');
var bodyParser = require('body-parser');
```

```
var db = require('./model/db'),
    speciality = require('./model/specialties'),
    group = require('./model/groups'),
    student = require('./model/students');

var app = express();
var ip = process.env.OPENSIFT_NODEJS_IP || '127.0.0.1',
    port = process.env.OPENSIFT_NODEJS_PORT || '8080';
app.listen(port, ip);
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

// uncomment after placing your favicon in /public
//app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());
var sessionConfig = {
    secret: 'KillerIsJim'
};
app.use(session(sessionConfig));
app.use(express.static(path.join(__dirname, 'public')));
var upload_dir = process.env.OPENSIFT_DATA_DIR;
if(!upload_dir) {
    upload_dir = path.join(__dirname, 'media');
}
app.use('/media', express.static(upload_dir));

require('./routes/index')(app);

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

```

var err = new Error('Not Found');
err.status = 404;
next(err);
});

// error handlers

// development error handler
// will print stacktrace
if (app.get('env') === 'development') {
  app.use(function(err, req, res, next) {
    res.status(err.status || 500);
    res.render('error', {
      message: err.message,
      error: err
    });
  });
}

```

```

// production error handler
// no stacktraces leaked to user
app.use(function(err, req, res, next) {
  res.status(err.status || 500);
  res.render('error', {
    message: err.message,
    error: {}
  });
});

```

```

module.exports = app;

```

block:index.js

```

/**
 * Created by tania on 6.02.16.
 */

```

```

function Block() {

}

Block.prototype.getTextForSkipLessons = function (n) {
    var result = n;
    if (n % 10 === 0) {
        result = n + ' пропущених пар';
    } else if (n % 10 === 1 && n !== 11) {
        result = n + ' пропущена пара';
    } else if ((n % 10 === 2 || n % 10 === 3 || n % 10 === 4) && (n !== 12 && n
    !== 13 && n !== 14)) {
        result = n + ' пропущені пари';
    } else {
        result = n + ' пропущених пар';
    }
    return result;
}

Block.prototype.getStudentPhoto = function (student) {
    var photoPath = student.photo;
    if (photoPath) {
        photoPath = '/media/' + photoPath;
    } else {
        if (student.male) {
            photoPath = '/images/person.png';
        } else {
            photoPath = '/images/Female.png';
        }
    }
    return photoPath;
}

Block.prototype.getTextForResponsible = function(male){
    var text = 'Відповідальний! ';

```

```
    if(!male){
        text = 'Відповідальна! '
    }
    return text;
};

Block.prototype.getTextForTalent = function(male){
    var text = 'Талановитий! ';
    if(!male){
        text = 'Талановита! '
    }
    return text;
};

Block.prototype.getTextForPositive = function(male){
    var text = 'Позитивний! ';
    if(!male){
        text = 'Позитивна! '
    }
    return text;
};

Block.prototype.getTextForCharisma = function(male){
    var text = 'Харизматичний! ';
    if(!male){
        text = 'Харизматична! '
    }
    return text;
};

Block.prototype.getTextForArsehole = function(male){
    var text = 'Намагався сподобатися викладачу! ';
    if(!male){
        text = 'Намагалась сподобатися викладачу! '
    }
    return text;
};

module.exports = new Block();
```

helper: color.js

```

/**
 * Created by tania on 7.02.16.
 */
function Color(){

}

Color.prototype.getRandomColor = function() {
  var letters = '0123456789ABCDEF'.split('');
  var color = '#';
  for (var i = 0; i < 6; i++ ) {
    color += letters[Math.floor(Math.random() * 16)];
  }
  return color;
}

module.exports = new Color()
                                middleware:checkAdmin.js
var assert = require('assert');

```

```

module.exports = function (req, res, next) {
  if (!req.session.user && req.originalUrl !== '/admin/login') {
    return res.redirect('/admin/login');
  }
  return next();
};

```

model:db.js

```

var mongoose = require('mongoose');
var mongoHost = process.env.OPENSIFT_MONGODB_DB_HOST ||
'localhost';
var mongoPort = process.env.OPENSIFT_MONGODB_DB_PORT ||
'28017';
var dbName = process.env.OPENSIFT_APP_NAME || 'halloffame';
var userName = process.env.OPENSIFT_MONGODB_DB_USERNAME;
var password = process.env.OPENSIFT_MONGODB_DB_PASSWORD;
if(userName){

```



```

    mongoose.connect('mongodb://' + userName + ':' + password + '@' +
mongoHost + ':' + mongoPort + '/' + dbName);
} else {
    mongoose.connect('mongodb://' + mongoHost + '/' + dbName);
}

```

model:groups.js

```
/**
```

```
* Created by tania on 29.01.16.
```

```
*/
```

```
var mongoose = require('mongoose');
```

```
var groupsSchema = new mongoose.Schema({
```

```
  title: {type: String, min: 1, max: 100, required: true, unique: true},
```

```
  year_start: {type: Number, required: true},
```

```
  _speciality: {type: mongoose.Schema.ObjectId, ref: 'Speciality'}
```

```
});
```

```
mongoose.model('Group', groupsSchema);
```

model:specialties.js

```
/**
```

```
* Created by tania on 6.02.16.
```

```
*/
```

```
var mongoose = require('mongoose');
```

```
var specialitySchema = new mongoose.Schema({
```

```
  title: {type: String, min: 1, max: 1000, required: true, unique: true},
```

```
  created_at: {type: Date, default: Date.now}
```

```
});
```

```
mongoose.model('Speciality', specialitySchema);
```

model:students.js

```
/**
```

```
* Created by tania on 29.01.16.
```

```
*/
```

```
var mongoose = require('mongoose');
```

```
var studentsSchema = new mongoose.Schema({
```

```
  name: String,
```

```
  photo: String,
```

```
  headman: Boolean,
```

```

n: Number,
group: {type: mongoose.Schema.ObjectId, ref: 'Group'},
link: String,
male: Boolean,
badges:{
  responsible:Boolean,
  talent: Boolean,
  positive: Boolean,
  charisma: Boolean,
  more: Boolean,
  arsehole: Boolean
}
});
mongoose.model('Student', studentsSchema);
public:stylesheets: style.css

```

```

body {
  overflow: auto;
  min-width: 765px;
}
h1, h2, h3, h4, h5, h6 {
  font-family: 'PT Serif', 'PT Serif1', serif !important;
  font-weight: bold !important;
}
#barChart,#chart-area{
  padding: 10px;
  max-width: 400px;
  max-height: 400px;
}
body {
  background-color: #ccc;
  font-family: 'Roboto', sans-serif !important;
}
.course{
  border-bottom: 5px dashed white;
}

```

```
}  
.top-menu {  
  text-align: center;  
}  
.img.headman {  
  width: 100%;  
  text-align: center;  
  overflow: hidden;  
}  
.top-section {  
  background-color: white;  
  width: 100%;  
}  
h1.top-name {  
  /*text-transform: uppercase;*/  
  text-align: center;  
  font-weight: 700;  
  font-size: 3rem;  
  margin: 0;  
}  
@media only screen and (min-width : 1100px) {  
  .container {  
    width: 98% !important;  
  }  
}  
@media only screen and (max-width : 960px) {  
  .students-items.row > .col.s12.m2 {  
    width: 25%;  
  }  
}  
@media only screen and (max-width : 640px) {  
  .students-items.row > .col.s12.m2 {  
    width: 50%;  
  }  
}
```

```
.students-items > *{
  display: flex;
  margin-left: 0 !important;
}

.students-items {
  display: flex;
  -webkit-flex-wrap: wrap;
  -moz-flex-wrap: wrap;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
}

.students-item-wrapper{
  margin-bottom: 5px;
  margin-left: 0 !important;
  padding: 0 5px !important;
}

.students-items .card{
  width: 100%;
}

.students-item {
  background-size: cover;
  position: relative;
  overflow: hidden;
  top: 0;
  left: 0;
  bottom: 0;
  right: 0;
  color: white;
  text-align: center;
  margin-left: 0 !important;
}

.students-item::before{
  content: "";
  display: block;
```

```
padding-top: 100% !important;
}
.card .card-image div.photo{
position: absolute !important;
top: 0;
right: 0;
left: 0;
bottom: 0;
}
.students-item-info {
position: relative !important;
z-index: 1000;
line-height: 1.3;
padding: 1px 5px;
text-align: center;
background-color: white;
color: #666 !important;
padding-bottom: 5px;
}
.card > a:hover{
text-decoration: underline;
color: #000000 !important;
box-shadow: 0 0 20px rgba(0,0,0,0.4) !important; /* Параметры тени */
}
.card > a:hover .students-item-info{
color: #000000 !important;
}
.students-item-name {
font-weight: 700;
text-transform: uppercase;
margin-top: 0.3rem;
}
.card .student-card div.label{
```

```
width: 30px;
height: 30px;
display: inline-block ;
border-radius: 50%;
margin-left: 5px;
vertical-align: top;
}

.card .student-card div.headman{
  background: url("../images/headman.jpg");
  background-size: 34px 34px;
  background-position: -2px -2px;

}

.card .student-card div.n0{
  background: url("../images/0n.jpg");
  background-size: 34px 34px;
  background-position: -2px -2px;
}

.card .student-card div.n20{
  background: url("../images/20n.jpg");
  background-size: 34px 34px;
  background-position: -2px -2px;
}

.card .student-card div.responsible{
  background: url("../images/badges/responsible.jpg");
  background-size: 34px 34px;
  background-position: -2px -2px;
}

.card .student-card div.talent{
  background: url("../images/badges/talent.jpg");
  background-size: 34px 34px;
  background-position: -2px -2px;
}
```

```
.card .student-card div.positive{  
  background: url("../images/badges/positive.jpg");  
  background-size: 34px 34px;  
  background-position: -2px -2px;  
}
```

```
.card .student-card div.charisma{  
  background: url("../images/badges/charisma.jpg");  
  background-size: 34px 34px;  
  background-position: -2px -2px;  
}
```

```
.card .student-card div.more{  
  background: url("../images/badges/more.jpg");  
  background-size: 34px 34px;  
  background-position: -2px -2px;  
}
```

```
.card .student-card div.arsehole{  
  background: url("../images/badges/arsehole.jpg");  
  background-size: 34px 34px;  
  background-position: -2px -2px;  
}
```

```
.students-item-description{  
  color: #000000;  
}
```

```
.student-card{  
  position: relative;  
}
```

```
.cos-show-more {  
  height: 4rem;  
  border: 0.2rem solid #222222;  
  box-sizing: border-box;
```

```
padding: 0 3rem;
text-transform: uppercase;
background: transparent;
margin: 5rem auto 6rem;
display: block;
border-radius: 2rem;
outline: none;
font-weight: 500;
font-size: 1.1rem;
color: #222222
}
.black-line{
width: 100%;
height: 5px;
background-color: #000000;
}
.top-section{
background-color: white;
}
.valign-wrapper-bottom{
display: -webkit-flex;
display: -ms-flexbox;
display: flex;
-webkit-align-items: center;
-ms-flex-align: center;
align-items: flex-end;
}
.valign-wrapper-bottom .valign{
display: block;
}
.top-section .row{
margin: 0 auto;
}
.top-section .row .col{
margin: 0 auto;
```



```

}
.top-section .description{
  color: #777;
}
.leaderbord .navbar-fixed,.menu-smint{
  z-index: 100000;
  min-width: 765px;
}
.top-label-img{
  width: 100%;
}
.good-label{
  color: green;
}
.bad-label{
  color: #ff0000;
}
.teacher-label{
  color: #0000ff;
  font-size: 9px;
  padding: 0 1px;
}
.teacher-label.bad-label{
  color: #ff0000;
}
.header-row > *{
  padding: 0 !important;
  position: relative;
}
#barLegend {
  text-align: center;
  font-weight: bold;
}
#barLegend span:first-child{
  margin-left: 10px;
}

```

```

    height: 10px;
    width: 10px;
    display: inline-block;
    min-width: 10px;
    min-height: 10px;
    margin-right: 10px;
}
#barLegend span:last-child{
    display: inline-block;
    font-weight: bold;
    line-height: 14px;
}
.label-wrapper{
    position: absolute;
    display: flex;
    flex-direction: column;
    z-index: 10000;
}
.label-wrapper .label{
    margin: 1px;
}

```

routes:admin:groups:edit.js

```

/**
 * Created by tania on 30.01.16.
 */
var mongoose = require('mongoose');
var async = require('async');
module.exports.get = function (req, res, next) {
    var groupId = req.params.groupId;
    async.parallel({
        group: function(callback){
            mongoose.model('Group').findById(groupId, function(err, group){
                callback(err, group);
            })
        },
    },

```

```

specialties: function(callback){
    mongoose.model('Specialty').find({}, function(err, specialties){
        callback(err, specialties)
    })
}
}, function(err, result){
    if(err) return next(err);
    return res.render('admin/groups/edit', {
        title: 'Express',
        group: result.group,
        specialties: result.specialties
    });

});

}

module.exports.post = function (req, res, next) {
    var groupId = req.params.groupId;
    mongoose.model('Group').findById(groupId, function (err, group) {
        if (err) {
            return next();
        } else {
            group.update({
                title: req.body.title,
                year_start: req.body.year_start,
                _speciality: req.body.specialties
            }, function (err, group) {
                if (err) {
                    return new Error('Помилка при збереженні курсу');
                }
                return res.redirect("/admin/groups");
            })
        }
    })
}
})
}

```

routes:admin:groups:index.js

```
/**
 * Created by tania on 29.01.16.
 */
module.exports = function (app) {
  app.get('/admin/groups', require('./list').get);

  app.get('/admin/groups/new', require('./new').get);
  app.post('/admin/groups/new', require('./new').post);

  app.get('/admin/groups/:groupId/edit', require('./edit').get);
  app.post('/admin/groups/:groupId/edit', require('./edit').post);
}
```

routes:admin:groups:list.js

```
/**
 * Created by tania on 29.01.16.
 */
var mongoose = require('mongoose');
module.exports.get = function (req, res, next) {
  mongoose.model('Group').find({}).populate('_speciality').exec(function
(err, groups) {
    if (err) {
      return next(new Error('Якась дивна помилка'));
    }
    console.log(groups);
    return res.render('admin/groups/list', {
      title: 'Group | Admin',
      groups: groups
    });
  });
};
```

routes:admin:groups:new.js

```
/**
 * Created by tania on 29.01.16.
 */
```

```

var mongoose = require('mongoose');
module.exports.get = function (req, res, next) {
  mongoose.model('Speciality').find({}).exec(function (err, specialties) {
    if (err) {
      return next(new Error("Помилка при пошуку, всіх спеціальностей..."))
    }
    return res.render('admin/groups/new', {
      title: 'Створення групи',
      specialties: specialties
    });
  });
}
module.exports.post = function (req, res, next) {
  console.log(req.body.specialties);
  mongoose.model('Group').create({
    title: req.body.title,
    year_start: req.body.year_start,
    _speciality: req.body.specialties
  }, function (err, group) {
    console.log(err);
    if (err) {
      return next(new Error('asd'));
    }
    return res.redirect('/admin/groups');
  })
};

```

routers:admin:login:index.js

/**

** Created by tania on 2.02.16.*

*/

```

module.exports = function (app) {
  app.get('/admin/login', require('./login').get);
  app.post('/admin/login', require('./login').post);
}

```

```

}

                                routes:admin:login:login.js

/**
 * Created by tania on 2.02.16.
 */
var user = require('.././../config').user;
module.exports.get = function (req, res, next) {
  return res.render('admin/login/login', {title: 'Express'});
}
module.exports.post = function(req,res,next){
  if (user.login === req.body.login && user.password ===
req.body.password) {
    req.session.user = {
      login: req.body.login,
      password: user.password
    }
  }
  console.log("USER1");
  console.log(req.session.user);
  res.redirect('/admin');
}

```

```

                                routes:admin:specialties:index.js

/**
 * Created by tania on 29.01.16.
 */
module.exports = function (app) {
  app.get('/admin/specialties', require('./list').get);

  app.get('/admin/specialties/new', require('./new').get);
  app.post('/admin/specialties/new', require('./new').post);
}

```

```

                                routes:admin:specialties:list.js

/**
 * Created by tania on 29.01.16.
 */

```

```

var mongoose = require('mongoose');
module.exports.get = function (req, res, next) {
  mongoose.model('Speciality').find({}).sort({created_at: -1}).exec(function
(err, specialties) {
  if (err) {
    return next(new Error('Якась дивна помилка'));
  }
  return res.render('admin/specialties/list', {
    title: 'specialties | Admin',
    specialties: specialties
  });
});
};

```

routes:admin:specialties:new.js

```

/**
 * Created by tania on 29.01.16.
 */
var mongoose = require('mongoose');
module.exports.get = function (req, res, next) {
  res.render('admin/specialties/new', {title: 'Express'});
}
module.exports.post = function (req, res, next) {
  console.log(req.body.title);
  mongoose.model('Speciality').create({
    title: req.body.title
  }, function (err, speciality) {
    if (err) {
      return next(new Error('asd'));
    }
    return res.redirect('/admin/specialties');
  })
};

```

routes:admin:students:delete.js

```

/**
 * Created by tania on 30.01.16.
 */
var mongoose = require('mongoose');
var async = require('async');

module.exports.get = function (req, res, next) {
  var studentId = req.params.studentId;
  mongoose.model('Student').findById(studentId, function (err, student) {
    if(err || !student) {return next()}
    else{
      student.remove(function (err, student){
        return res.redirect('/admin/students');
      });
    }
  })
}

```

routes:admin:students:edit.js

```

/**
 * Created by tania on 30.01.16.
 */
var mongoose = require('mongoose');
var async = require('async');
module.exports.get = function (req, res, next) {
  async.parallel({
    groups: function(callback){
      mongoose.model('Group')
        .find({})
        .exec(function(err,groups){
          callback(err,groups);
        })
    },
    student: function(callback){

```



```

    var studentId = req.params.studentId;
    if(!studentId) {callback(new Error(),null)}
    else{

mongoose.model('Student').findById(studentId,function(err,student){
    callback(err,student);
    });
    }
    },function(err,result){
    if(err){
        return next(err);
    }else if(!result.student){
        return next();
    }else{
        res.render('admin/students/edit', {
            title: result.student.name + 'Edit | Admin',
            student: result.student,
            groups: result.groups
        });
    }
    });
}

```

```

var upload_dir = process.env.OPENSIFT_DATA_DIR || 'media';
var multer = require('multer');
var storage = multer.diskStorage({
    destination: function (req, file, callback) {
        callback(null, upload_dir);
    },
    filename: function (req, file, callback) {
        callback(null, Date.now() + file.originalname);
    }
});
var upload = multer({storage: storage}).single('photo');

```

```
module.exports.post = function (req, res, next) {  
  var studentId = req.params.studentId;  
  mongoose.model('Student').findById(studentId, function (err, student) {  
    if(err || !student) {return next()}  
    else{  
      upload(req,res,function(err){  
        if(err){  
          return next(new Error('Не вийшло завантажити  
фотографію'));  
        }else{  
          var file = req.file;  
          var student1 = {  
            name: req.body.name,  
            headman: req.body.headman,  
            n: req.body.n,  
            group: req.body.group,  
            link: req.body.link,  
            male: req.body.male,  
            "badges.responsible" : req.body.responsible,  
            "badges.talent" : req.body.talent,  
            "badges.positive" : req.body.positive,  
            "badges.charisma" : req.body.charisma,  
            "badges.more" : req.body.more,  
            "badges.arsehole" : req.body.arsehole  
          }  
          if (file) {  
            student1.photo = file.filename;  
          };  
          student.update(  
            student1,  
            function(err,student){  
              if(err){  
                return next(new Error('Не вдалось створити  
студента')));  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```

        }
        return res.redirect('/admin/students');
    }
)

}
});
}
})
}

```

routes:admin:students:index.js

```

/**
 * Created by tania on 29.01.16.
 */
module.exports = function (app) {
    app.get('/admin/students', require('./list').get);

    app.get('/admin/students/new', require('./new').get);
    app.post('/admin/students/new', require('./new').post);

    app.get('/admin/students/:studentId/edit', require('./edit').get);
    app.post('/admin/students/:studentId/edit', require('./edit').post);

    app.get('/admin/students/:studentId/delete', require('./delete').get);
}

```

routes:admin:students:list.js

```

/**
 * Created by tania on 29.01.16.
 */
var mongoose = require('mongoose');
module.exports.get = function (req, res, next) {
    mongoose.model('Student').find({})
        .populate('group')
        .exec(function (err, students) {
            if (err) {

```

```

        return next(new Error('Якась дивна помилка'));
    }
    return res.render('admin/students/list', {
        title: 'Students | Admin',
        students: students
    });
});

};

                                routes:admin:students:new.js

/**
 * Created by tania on 29.01.16.
 */

var upload_dir = process.env.OPENSIFT_DATA_DIR || 'media';
var multer = require('multer');
var storage = multer.diskStorage({
    destination: function (req, file, callback) {
        callback(null, upload_dir);
    },
    filename: function (req, file, callback) {
        callback(null, Date.now() + file.originalname);
    }
});
var upload = multer({storage: storage}).single('photo');

var mongoose = require('mongoose');
module.exports.get = function (req, res, next) {
    mongoose.model('Group').find({}, function(err, groups){
        if(err){
            return next(new Error('Помилка при завантаженні груп'));
        }
        res.render('admin/students/new', {
            title: 'Express',
            groups: groups
        });
    });
});

```

```

};
module.exports.post = function (req, res, next) {
  console.log('POST');
  upload(req,res,function(err){
    if(err){
      return next(new Error('Не вийшло завантажити фотографію'));
    }else{
      var file = req.file;
      var student = {
        name: req.body.name,
        headman: req.body.headman,
        n: req.body.n,
        group: req.body.group,
        link: req.body.link,
        male: req.body.male
      }
      if (file) {
        student.photo = file.filename;
      };
      mongoose.model('Student').create(
        student,
        function(err,student){
          if(err){
            return next(new Error('Не вдалось створити студента'));
          }
          return res.redirect('/admin');
        }
      )
    }
  });
};

```

routes:admin:index.js

/**

** Created by tania on 29.01.16.*

```

*/
var checkAuthorize = require('../middleware/checkAdmin');
module.exports = function(app) {
  app.get('/admin/*', checkAuthorize, function(req, res, next){
    next();
  });
  app.post('/admin/*', checkAuthorize, function(req, res, next){
    next();
  });
  app.get('/admin', checkAuthorize, function(req, res, next) {
    res.render('admin/index', { title: 'Admin' });
  });
  require('./groups')(app);
  require('./students')(app);
  require('./specialties')(app);
  require('./login')(app);
}

```

routes:index.js

```

var mongoose = require('mongoose');
var async = require('async');
var _ = require('underscore');
var colorHelper = require('../helper/color');
module.exports = function (app) {
  app.get('/', function (req, res, next) {
    mongoose.model('Student').find({})
      .sort({
        n: 1,
        headman: -1
      })
      .populate('group ')
      .exec(function (err, students) {
        if (err) {
          return next();
        }
        var Speciality = mongoose.model('Speciality');

```

```

var Group = mongoose.model('Group');
Group.populate(students, {
  path: 'group._speciality',
  select: 'title',
  model: Speciality
}, function(err, students) {
  if (err) {
    return next();
  }
  var fullStudent = students.length;
  var speciality = _.groupBy(students, function(student) {
    return student.group._speciality ? student.group._speciality.title
: null;
  });
  speciality = _.map(speciality, function(value, key) {
    return {
      fillColor: colorHelper.getRandomColor(),
      data:[value.length],
      label: key
    }
  });
  var students = _.groupBy(students, function (student) {
    return (new Date().getFullYear() - student.group.year_start)
  });
  var studentCount = _.map(students, function(value, key) {
    return {
      value:value.length,
      color: colorHelper.getRandomColor(),
      label: key + ' курс'
    }
  });
  return res.render('index', {
    title: 'КРАЩІ СТУДЕНТИ ЖДТУ ФІКТ',
    students: students,
    block: require('../block/index'),
  });
});

```

```

        chart: JSON.stringify(studentCount),
        fullStudentCount: fullStudent,
        sessionId: req.sessionID,
        speciality: JSON.stringify(speciality)
    });
});
});
});
require('./admin/index')(app);
}

view:admin:groups:edit.jade

extends ../layout
append content
    div.box.box-info
        div.box-header.with-border
            h3.box-title Редагувати групи

form#formAddCourse.form-horizontal(method="post",action="/admin/groups/#{group._id}/edit")
    div.box-body
        div.form-group
            label.col-sm-2.control-label(for="inputTitle") Назва групи
            div.col-sm-10

input#inputTitle.form-control(value="#{group.title}",type="text",
name="title",autocomplete="off")
    div.box-body
        div.form-group
            label.col-sm-2.control-label(for="inputStartYear") Рік вступу
            div.col-sm-10

input#inputStartYear.form-control(value="#{group.year_start}",type="text",
name="year_start",autocomplete="off")
    div.box-body
        div.form-group

```



```

label.col-sm-2.control-label(for="inputSpec") Спеціальність
div.col-sm-10
  select#inputSpec.form-control( name="specialties")
    - each specialty, i in specialties
      if(specialty._id.equals(group._speciality))
        option(value='#{specialty._id}',selected)
          #{specialty.title}
      else
        option(value='#{specialty._id}') #{specialty.title}
  div.box-footer
    button#btnSubmit.btn.btn-primary.pull-right(type="submit")
Зберегти зміни
views:admin:groups:list.jade
extends ../layout
append content
div.box.admin-courses-index
  div.box-header
    h1.box-title #{title}
    a.btn.btn-info.pull-right(href="/admin/groups/new") Додати групу
  div.box-body
    table.table.table-bordered.table-hover.dataTable(role="grid")
      thead
        tr.row
          th Назва
          th Спеціальність
          th Рік початку навчання
      tbody
        - each group, i in groups
          tr.row
            th #{group.title}
            th
              if(group._speciality)
                span #{group._speciality.title}
            th #{group.year_start}
            th

```

```



```

```

extends ../layout
append style
block body
  div.login-box
    div.login-logo Адмінка
    div.login-box-body
      p.login-box-msg Sign in to start your session
      div.login-box-msg
        form(action="/admin/login", method="post")
          div.form-group.has-feedback
            input.form-control(type="login",name="login",placeholder="Email")
              span.glyphicon.glyphicon-envelope.form-control-feedback
            div.form-group.has-feedback
              input.form-control(type="password",name="password",placeholder="Password")
                span.glyphicon.glyphicon-lock.form-control-feedback
            div.form-group
              button.btn.btn-primary.btn-block.btn-flat.right(type="submit")

```

Sign In

views:admin:specialties:list.jade

```

extends ../layout
append content
  div.box.admin-courses-index
    div.box-header
      h1.box-title #{title}
      a.btn.btn-info.pull-right(href="/admin/specialties/new") Додати спеціальність
    div.box-body
      table.table.table-bordered.table-hover.dataTable(role="grid")
        thead
          tr.row
            th Назва
        tbody

```

- each *speciality*, *i* in specialties

tr.row

th *#{speciality.title}*

th *#{speciality.year_start}*

th

div

a(*href*='/admin/speciality/*#{speciality._id}*/edit')

Редагувати

views:admin:specialties:new.jade

extends ../layout

append **content**

div.box.box-info

div.box-header.with-border

h3.box-title **Добавити нову спеціальність**

form#formAddCourse.form-horizontal(*method*="post",*action*="/admin/specialties/new")

div.box-body

div.form-group

label.col-sm-2.control-label(*for*="inputTitle") **Назва**

спеціальності

div.col-sm-10

input#inputTitle.form-control(*type*="text",
name="title",*autocomplete*="off")

div.box-footer

button#btnSubmit.btn.btn-primary.pull-right(*type*="submit")

Створити спеціальність

views:admin:students:edit.jade

extends ../layout

append **content**

div.box.box-info

div.box-header.with-border

h3.box-title **Редагувати** *#{student.name}*

```
form#formAddCourse.form-horizontal(method="post",enctype="multipart/
form-data",action="/admin/students/#{student._id}/edit")
```

```
div.box-body
```

```
div.form-group
```

```
label.col-sm-2.control-label(for="inputName") Ім'я
```

```
div.col-sm-10
```

```
input#inputName.form-control(value='#{student.name}',type="text",
name="name",autocomplete="off")
```

```
div.form-group
```

```
label.col-sm-2.control-label(for="inputPhoto") Фото
```

```
div.col-sm-1
```

```
img(src='/media/#{student.photo}',style='width:50px')
```

```
div.col-sm-9
```

```
input#inputPhoto.form-control(type="file",
name="photo",autocomplete="off")
```

```
div.form-group
```

```
label.col-sm-2.control-label(for="inputHeadman") Староста?
```

```
div.col-sm-10
```

```
if(student.headman)
```

```
input#inputHeadman(checked,type="checkbox",
name="headman",autocomplete="off")
```

```
else
```

```
input#inputHeadman(type="checkbox",
name="headman",autocomplete="off")
```

```
div.form-group
```

```
label.col-sm-2.control-label(for="inputN") Кількість Енок
```

```
div.col-sm-10
```

```
input#inputN.form-control(value='#{student.n}',type="text",
name="n",autocomplete="off")
```

```
div.form-group
```

```
label.col-sm-2.control-label(for="inputGroup") Група
```

```
div.col-sm-10
```

```

select#inputGroup.form-control( name="group")
- each group, i in groups
    if(group._id.equals(student.group))
        option(value='#{group._id}', selected) #{group.title}
    else
        option(value='#{group._id}') #{group.title}
div.form-group
    label.col-sm-2.control-label(for="inputLink") Лінка
div.col-sm-10

input#inputLink.form-control(value='#{student.link}',type="text",
name="link",autocomplete="off")
div.form-group
    label.col-sm-2.control-label(for="inputMale") Чоловік?
div.col-sm-10
    if(student.male)
        input#inputMale(checked,type="checkbox",
name="male",autocomplete="off")
    else
        input#inputMale(type="checkbox",
name="male",autocomplete="off")
div.form-group
    label.col-sm-2.control-label(for="inputResponsible")
Відповідальний?
div.col-sm-10
    if(student.badges.responsible)
        input#inputMale(checked,type="checkbox",
name="responsible",autocomplete="off")
    else
        input#inputMale(type="checkbox",
name="responsible",autocomplete="off")
div.form-group
    label.col-sm-2.control-label(for="inputMale") Талановитий?
div.col-sm-10
    if(student.badges.talent)

```

```

        input#inputMale(checked,type="checkbox",
name="talent",autocomplete="off")
    else
        input#inputMale(type="checkbox",
name="talent",autocomplete="off")
    div.form-group
        label.col-sm-2.control-label(for="inputMale") positive?
    div.col-sm-10
        if(student.badges.positive)
            input#inputMale(checked,type="checkbox",
name="positive",autocomplete="off")
        else
            input#inputMale(type="checkbox",
name="positive",autocomplete="off")
    div.form-group
        label.col-sm-2.control-label(for="inputMale") charisma?
    div.col-sm-10
        if(student.badges.charisma)
            input#inputMale(checked,type="checkbox",
name="charisma",autocomplete="off")
        else
            input#inputMale(type="checkbox",
name="charisma",autocomplete="off")
    div.form-group
        label.col-sm-2.control-label(for="inputMale") more?
    div.col-sm-10
        if(student.badges.more)
            input#inputMale(checked,type="checkbox",
name="more",autocomplete="off")
        else
            input#inputMale(type="checkbox",
name="more",autocomplete="off")
    div.form-group
        label.col-sm-2.control-label(for="inputMale") arsehole?
    div.col-sm-10

```

```
        if(student.badges.arsehole)
            input#inputMale(checked,type="checkbox",
name="arsehole",autocomplete="off")
        else
            input#inputMale(type="checkbox",
name="arsehole",autocomplete="off")
```

```
div.box-footer
    button#btnSubmit.btn.btn-primary.pull-right(type="submit")
Редагувати студента
```

```
views:admin:students:list.jade
extends ../layout
append content
div.box.admin-courses-index
div.box-header
    h1.box-title #{title}
    a.btn.btn-info.pull-right(href="/admin/students/new") Додати
групу
div.box-body
    table.table.table-bordered.table-hover.dataTable(role="grid")
        thead
            tr.row
                th Фото
                th Ім'я
                th Група
                th Відзнаки
                th Лінка
        tbody
            - each student, i in students
                tr.row
                    th
                        img(src='/media/#{student.photo}',style='width:50px')
                    th #{student.name}
                    th #{student.group.title}
```



```

th #{student.link}
th
  span Відповідальність #{student.badges.responsible}
  span talent #{student.badges.talent}
  span positive #{student.badges.positive}
  span charisma #{student.badges.charisma}
  span more #{student.badges.more}
  span arsehole #{student.badges.arsehole}

th
div
  a(href="/admin/students/#{student._id}/edit")

```

Редагувати

views:admin:students:new.jade

extends ../layout

append content

div.box.box-info

div.box-header.with-border

h3.box-title Додати нового студента

form#formAddCourse.form-horizontal(method="post",enctype="multipart/form-data",action="/admin/students/new")

div.box-body

div.form-group

label.col-sm-2.control-label(for="inputName") Ім'я

div.col-sm-10

input#inputName.form-control(type="text",name="name",autocomplete="off")

div.form-group

label.col-sm-2.control-label(for="inputPhoto") Фото

div.col-sm-10

input#inputPhoto.form-control(type="file",name="photo",autocomplete="off")

div.form-group

label.col-sm-2.control-label(for="inputHeadman") Староста?

```

    div.col-sm-10
      input#inputHeadman(type="checkbox",
name="headman",autocomplete="off")
    div.form-group
      label.col-sm-2.control-label(for="inputN") Кількість Енок
    div.col-sm-10
      input#inputN.form-control(type="text",
name="n",autocomplete="off")
    div.form-group
      label.col-sm-2.control-label(for="inputGroup") Група
    div.col-sm-10
      select#inputGroup.form-control( name="group")
        - each group, i in groups
          option(value='#{group._id}') #{group.title}
    div.form-group
      label.col-sm-2.control-label(for="inputLink") Лінка
    div.col-sm-10
      input#inputLink.form-control(type="text",
name="link",autocomplete="off")
    div.form-group
      label.col-sm-2.control-label(for="inputMale") Чоловік?
    div.col-sm-10
      input#inputMale(type="checkbox",
name="male",autocomplete="off")
    div.box-footer
      button#btnSubmit.btn.btn-primary.pull-right(type="submit")

```

Створити групу

views:admin:index.jade

extends layout

append content

h1 dashboard

views:admin:layout.jade

extends ../layout

block style

```

    link(rel='stylesheet',
href='/bower_components/AdminLTE/bootstrap/css/bootstrap.min.css')
    link(rel='stylesheet',
href='/bower_components/AdminLTE/dist/css/AdminLTE.min.css')
    link(rel='stylesheet',
href='/bower_components/AdminLTE/dist/css/skins/skin-blue.min.css')
    link(rel='stylesheet', href='/stylesheets/admin/style.css')
block header
block body
    block main-header
        header.main-header
    block aside
        aside.main-sidebar
            section.sidebar
                ul.sidebar-menu
                    li.header
                        a(href="/admin/groups") Группи
                    li.header
                        a(href="/admin/specialties") Спеціальність
                    li.header
                        a(href="/admin/students") Студенти
div.content-wrapper
    section.content
        block content
div.main-footer
    block main-footer
        div.pull-right.hidden-xs Anything you want
        strong All rights reserved.
block footer
append js
    script(type='text/javascript'
src="/bower_components/AdminLTE/bootstrap/js/bootstrap.min.js")
    script(type='text/javascript'
src="/bower_components/AdminLTE/dist/js/app.min.js")

```

views:error.jade

extends layout

block body

h1= message

h2= error.status

pre #{error.stack}

views:index.jade

extends layout

append style

link(href='/bower_components/Materialize/bin/materialize.css',
type="text/css", rel="stylesheet")

block body

div.leaderbord

div.navbar

nav.menu-smint.black

div.nav-wrapper.container.fixed

a.top-name.subNavBtn(href='#sectionTop') Кращі студенти
факультету ІКТ

sectionTeacher

ul.right

-for studentsOneGroup, i in students

li

a.subNavBtn(href='#section#{i}') #{ i } курс

div.top-section.sectionTop

h1.top-name Кращі студенти факультету ІКТ

div.row.valign-wrapper-bottom.header-row

div.col.s12.m4.center

canvas#barChart(height="450",width="600")

div#barLegend

div.col.s12.m4.valign

h3.center.description.valign Студенти, що мають середній бал
5.0 за другу сесію 2015-2016р

div.center

```



```

```

    -for student, i in studentsOneGroup
      div.col.s12.m2
        div.card
          -var div = student.link ? "a" : "div"

      #{div}.student-card(href='#{student.link}',target='_blank')
        div.label-wrapper
          if(student.headman)
            div(title='Сtrapocтa').headman.label
          if(student.n == 0)

        div(title='#{block.getTextForSkipLessons(student.n)}').n0.label
          else if(student.n > 19)

        div(title='#{block.getTextForSkipLessons(student.n)}').n20.label
          if(student.badges.responsible)

        div(title='#{block.getTextForResponsible(student.male)}').responsible.label
          if(student.badges.talent)

        div(title='#{block.getTextForTalent(student.male)}').talent.label
          if(student.badges.positive)

        div(title='#{block.getTextForPositive(student.male)}').positive.label
          if(student.badges.charisma)

        div(title='#{block.getTextForCharisma(student.male)}').charisma.label
          if(student.badges.more)
            div(title='Потрібно більше таких
студентів!').more.label
          if(student.badges.arsehole)

        div(title='#{block.getTextForArsehole(student.male)}').arsehole.label
          div.card-image
            div.students-item

```

```


photo(style="background:url('#{block.getStudentPhoto(student)}')
no-repeat center top / cover")


black-line

students-item-info

students-item-name #{student.name}


students-item-description Грына
#{student.group.title+' '}


classs = (student.n == 0) ? 'good-label' :
(student.n > 19 ? 'bad-label' : null);
span(class='#{classs}')
#{block.getTextForSkipLessons(student.n)}
if(student.headman)


good-label Стара


```

```

append js
script(type='text/javascript'
src="/bower_components/jquery/dist/jquery.js")
script(type='text/javascript' src="/vendor/smint/js/jquery.smint.js")
script(type='text/javascript').
$(document).ready( function() {
    $('.menu-smint').smint({
        'scrollSpeed' : 500
    });
});
script(type='text/javascript'
src="/bower_components/Chart.js/Chart.min.js")
script(type='text/javascript').
(function (i, s, o, g, r, a, m) {
    i['GoogleAnalyticsObject'] = r;
    i[r] = i[r] || function () {
        (i[r].q = i[r].q || []).push(arguments)
    }, i[r].l = 1 * new Date();

```

```

    a = s.createElement(o),
        m = s.getElementsByTagName(o)[0];
    a.async = 1;
    a.src = g;
    m.parentNode.insertBefore(a, m)
})(window, document, 'script',
'//www.google-analytics.com/analytics.js', 'ga');
ga('create', 'UA-73418436-1', 'auto');
ga('set', 'userId', '#{sessionId}');
ga('send', 'pageview');

script(type='text/javascript').
    var pieData = JSON.stringify(!{chart});
    pieData = JSON.parse(pieData)

    var datasets = JSON.stringify(!{speciality});
    datasets = JSON.parse(datasets);
    var barChartData = {
        labels: ["Остання сесія"],
        datasets: datasets
    }
    window.onload = function () {
        var ctx =
document.getElementById("chart-area").getContext("2d");
        window.myPie = new Chart(ctx).Pie(pieData,{
            tooltipEvents: [],
            onAnimationComplete: function() {
                this.showTooltip(this.segments, true);
            },
            tooltipTemplate: "<%= label %> : <%= (value * 100 /
#{fullStudentCount}).toFixed(2)*1 %> %",
            responsive: true
        });

```



```

var ctx1 =
document.getElementById("barChart").getContext("2d");
window.myBar = new Chart(ctx1).Bar(barChartData, {
    barShowStroke: true,
    responsive: true,
    legendTemplate: "<ul
class=\"<%=name.toLowerCase()%>-legend\"><%= for (var i=0;
i<datasets.length; i++){%><li><span
style=\"background-color:<%=datasets[i].fillColor%>\"></span><%=if(dat
asets[i].label){%><%=datasets[i].label%><%=}%></li><%=}%></ul>"

});
$('#barLegend').append(window.myBar.generateLegend());

}
script(type='text/javascript').
var reformalOptions = {
    project_id: 972005,
    project_host: "halloffame.reformal.ru",
    tab_orientation: "right",
    tab_indent: "50%",
    tab_bg_color: "#292421",
    tab_border_color: "#FFFFFF",
    tab_image_url:
"http://tab.reformal.ru/T9GC0LfRi9Cy0Ysg0Lgg0L%252FRgNC10LTQu
9C%252B0LbQtdC90LjRjw==/FFFFFF/a08a7c60392f68cb33f77d4f56cf8c
6f/right/1/tab.png",
    tab_border_width: 2
};

(function() {
var script = document.createElement('script');
script.type = 'text/javascript'; script.async = true;
script.src = ('https:' == document.location.protocol ? 'https://' :
'http://') + 'media.reformal.ru/widgets/v3/reformal.js';

```

```
document.getElementsByTagName('head')[0].appendChild(script);
})();
```

views:layout.jade

doctype html

html

head

title= title

meta(name="verify-reformal", content="81679bf86b2c3c600bdf7b9")

link(rel='stylesheet',

**href='https://fonts.googleapis.com/css?family=PT+Serif:700&subset=cyrilli
c-ext',type='text/css')**

link(rel='stylesheet',

href='https://fonts.googleapis.com/css?family=Roboto:500',type='text/css')

link(rel='stylesheet', href='/stylesheets/style.css')

block style

body.p-profile.hold-transition.skin-blue.sidebar-mini

block body

block js