```
/** * Deal Controller * @author Ashish T * @created 21 Nov 2016 * @version 1.0 */ 'use strict'; var self = this; var version = __api_version__; var Flip = __rootRequire('app/models/deal/deal_flip_model'); var Rental = __rootRequire('app/models/deal/deal_rental_model'); var Skulist = __rootRequire('app/models/deal/sku_list_model'); var propertyObj = __rootRequire('app/models/properties/properties.js'); var EmailTemplateObj = __rootRequire('app/models/templates/template.js'); var rh = __rootRequire('app/helpers/request_handler'); var c = __rootRequire('app/helpers/cypher'); var jwt = require('jsonwebtoken'); var request = require('request'); var async = require('async'); var config = __rootRequire('app/config/config'); var constantObj = __rootRequire('constants'); var lodash = require('lodash'); var pdf = require('html-pdf'); var fs = require('fs'); var nodemailer = require('nodemailer'); var config = __rootRequire('app/config/smtp'); var forEach = require('async-foreach').forEach; module.exports = { save_flip: save_flip, getFlipOutput: getFlipOutput, save_rental: save_rental, pmt: pmt, total_cash_outlay: total_cash_outlay, getRentalOutput: getRentalOutput, generateFlipTemplate: generateFlipTemplate, replace: replace, send_email: send_email, send_rental_email: send_rental_email } function generateFlipTemplate(req, res) { try { propertyObj.findOne({_id: req.body.property_id}).select({'address': 1, 'street_number': 1}).exec(function(err, propertyResult) { if (err) { return res.send({status: constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); } else { EmailTemplateObj.findOne({_id: '58a2fa01ebb782382d85b7a2'}).exec(function(err, lomEmailTemplate) { if (err) { return res.send({status: constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); } else { var baseUrl = req.protocol + '://' + req.get('host'); var options = {template: lomEmailTemplate.email_template_content, replacement: {" {{logo_url}}": baseUrl + "/assets/images/black-logo.png", " {{copyright}}": (new Date().getFullYear()) + " Copyright 2016", " {{link.abuse_email}}": "info@resimpli.com", "{{property_address}}": String(propertyResult.address)}}; var emailTemplate = replace(options.template.toString(), options.replacement); return res.send({status: constantObj.statusCode.success, data: emailTemplate}); } }); } }); } catch (err) { return res.json({status: 'ERROR', message: i18n.__('ERROR')}); } } function replace(str, replacement) { var re = new RegExp(Object.keys(replacement).join("|"), "gi"); str = str.replace(re, function(matched) { return replacement[matched.toLowerCase()]; });
```

```
return str; } /** * Send Email */ function send_email(req, res) { var commaNumber = require('comma-number') try { var FlipProjection = { created: false, is_deleted: false, __v: false }; Flip.findOne({users_id: req.user._id, properties_id: req.body.property_id}, FlipProjection) .exec(function(err, flipList) { if (err) { res.json({status: constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); } else { if (flipList) { var date = new Date(); var newDate = date.getTime(); var filePath = './public/uploads/flip/' + newDate + '.pdf'; var options = {border: { "top": "30px", // default is 0, units: mm, cm, in, px "right": "20px", "bottom": "30px", "left": "20px" }}; var pdfFormData = '
MAXIMUM PURCHASE PRICE CALCULATION
'; pdfFormData += '
Resale Price' + commaNumber(flipList.resale_price) + '
'; pdfFormData += '
Minimum Required Profit' + commaNumber(flipList.minimum_required_profit) + '
'; pdfFormData += '
Repair Costs' + commaNumber(flipList.repair_cost) + '
'; pdfFormData += '
Purchase Closing Cost' + commaNumber(flipList.purchase_closing_costs) + '
'; pdfFormData += '
Appraisal Fee' + commaNumber(flipList.appraisal_rate) + '
'; pdfFormData += '
Home Inspection Fee' + commaNumber(flipList.home_inspection_fee) + '
'; pdfFormData += '
Lender Costs
'; pdfFormData += '
Loan Interest' + commaNumber(flipList.loan_interest) + '
'; pdfFormData += '
Lender Points' + commaNumber(flipList.lender_points) + '
'; pdfFormData += '
Lender Fees' + commaNumber(flipList.lender_fees) + '
'; pdfFormData += '
Holding Costs
'; pdfFormData += '
Property Taxes' + commaNumber(flipList.property_taxes * flipList.project_duration) + '
'; pdfFormData += '
Electricity' + commaNumber(flipList.electricity * flipList.project_duration) + '
'; pdfFormData += '
```

```
Gas' + commaNumber(flipList.gas * flipList.project_duration) + '
'; pdfFormData += '
Water' + commaNumber(flipList.water * flipList.project_duration) + '
'; pdfFormData += '
Garbage' + commaNumber(flipList.garbage * flipList.project_duration)
+ '
'; pdfFormData += '
HOA' + commaNumber(flipList.hoa * flipList.project_duration) + '
'; pdfFormData += '
property_insurance' + commaNumber(flipList.property_insurance *
flipList.project_duration) + '
'; pdfFormData += '
Lawn Care/Snow Removal' + commaNumber(flipList.lawn_care *
flipList.project_duration) + '
'; pdfFormData += '
Other Holding Cost' + commaNumber(flipList.other_holding_costs *
flipList.project_duration) + '
'; pdfFormData += '
Other Holding Cost' + commaNumber(flipList.other_holding_costs *
flipList.project_duration) + '
'; pdfFormData += '
Selling Cost>
'; pdfFormData += '
Commission' + commaNumber(flipList.comm) + '
'; pdfFormData += '
Sales Closing Costs' + commaNumber(flipList.closing_costs) + '
'; pdfFormData += '
Closing Costs Credit to Buyer ' +
commaNumber(flipList.Closing_costs_credit_to_buyer) + '
'; pdfFormData += '
Commission' + commaNumber(flipList.comm) + '
'; pdfFormData += '
Maximum Purchase Price' +
commaNumber(flipList.maximum_purchase_price.toString().replace('-
', '')) + '
'; pdfFormData += '
PERSONAL FUNDS NEEDED
'; pdfFormData += '
Purchase Price' +
commaNumber(flipList.maximum_purchase_price.toString().replace('-
', '')) + '
'; pdfFormData += '
Repair Costs' + commaNumber(flipList.repair_cost) + '
'; pdfFormData += '
```

```
Purchase Closing Cost' +
commaNumber(flipList.purchase_closing_costs) + '
'; pdfFormData += '
Appraisal Fee' + commaNumber(flipList.appraisal_rate) + '
'; pdfFormData += '
Home Inspection Fee' + commaNumber(flipList.home_inspection_fee)
+ '
'; pdfFormData += '
Other Fee' + commaNumber(flipList.other_fee) + '
'; pdfFormData += '
Total Funds Needed' +
commaNumber(flipList.total_funds_needed.toString().replace('-', '')) +
'
'; pdfFormData += '
Total Loan Amount' + commaNumber(flipList.total_loan_amount) + '
'; pdfFormData += '
Out of Pocket Funds/(Excess from Loan)' +
commaNumber(flipList.out_of_pocket.toString().replace('-', '')) + '
'; pdfFormData += '
FINANCIAL SUMMARY
'; pdfFormData += '
Resale Price after Fix Up' + commaNumber(flipList.resale_price) + '
'; pdfFormData += '
Purchase Price' +
commaNumber(flipList.maximum_purchase_price.toString().replace('-
', '')) + '
'; pdfFormData += '
Purchase Cost ' + commaNumber(flipList.purchase_costs) + '
'; pdfFormData += '
Repair Costs ' + commaNumber(flipList.repair_cost) + '
'; pdfFormData += '
Holding Costs ' + commaNumber(flipList.holding_cost) + '
'; pdfFormData += '
Lender Costs ' + commaNumber(flipList.lender_costs) + '
'; pdfFormData += '
Selling Costs ' + commaNumber(flipList.selling_cost) + '
'; pdfFormData += '
Net Profit ' + commaNumber(flipList.net_profite.toString().replace('-',
'')) + '
'; pdfFormData += '
ROI CALCULATION
'; pdfFormData += '
Project ROI ' + commaNumber(flipList.project_roi.toString().replace('-
', '')) + '
```

'; pdfFormData += '
Project IRR ' + commaNumber(flipList.project_irr) + '
'; pdfFormData += '
FINANCING COST
'; pdfFormData += '
Loan Amount' + commaNumber(flipList.total_loan_amount) + '
'; pdfFormData += '
Lender Points($)' + commaNumber(flipList.lender_points) + '
'; pdfFormData += '
Lender Fees' + commaNumber(flipList.lender_fees) + '
'; pdfFormData += '
Interest Rate(%)' + commaNumber(flipList.interest_rate) + '
'; pdfFormData += '
Project Duration (months) ' +
commaNumber(flipList.project_duration) + '
'; pdfFormData += '
Monthly Loan Payment ' +
commaNumber(flipList.monthly_loan_payment) + '
'; pdf.create(pdfFormData, options).toFile(filePath, function(err,
pdfinfo) { if (err) { console.log(err) } else { var transporter =
nodemailer.createTransport(config.development.connectionURL); var
baseUrl = req.protocol + '://' + req.get('host'); var options =
{template: req.body.content, replacement: {"{{user.name}}":
'Ashoishj', "{{logo_url}}": baseUrl + "/assets/images/black-logo.png",
"{{copyright}}": (new Date().getFullYear()) + " Copyright 2016", "
{{link.abuse_email}}": "info@resimpli.com"}}; var emailTemplate =
replace(options.template.toString(), options.replacement); var
message = { from: 'info@resimpli.com', to: req.body.to, cc:
req.body.cc, subject: req.body.subject, html: emailTemplate,
attachments: [{ filename: newDate + '.pdf', contentType:
'application/pdf', path: filePath }] }; transporter.sendMail(message,
function(error, info) { if (error) { return console.log(error); }
res.json({status: constantObj.statusCode.success, message:
constantObj.messages.sendEmail}); }); } }); } else { res.json({status:
constantObj.statusCode.error, message:
constantObj.messages.emptyCalculation}); } } }); } catch (err) {
res.json({status: constantObj.statusCode.error, message:
constantObj.messages.emptyData}); } } /** * Send Email */ function
send_rental_email(req, res) { var commaNumber = require('comma-
number'); try { var RentalProjection = { created: false, is_deleted:
false, __v: false }; Rental.findOne({users_id: req.user._id,
properties_id: req.body.property_id}, RentalProjection)
.exec(function(err, rentalList) { if (err) { return res.send({status:
constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); }

```
else { if (rentalList) { /** Year one calculation **/ var
YearOne_monthly_rental_income =
(parseInt(rentalList.monthly_rental_income) * 12); var
YearOne_other_monthly_income =
(parseInt(rentalList.other_monthly_income) * 12); var
YearOne_total_gross_monthly_income =
(YearOne_monthly_rental_income + YearOne_other_monthly_income);
/* Expences */ var YearOne_vacancy_expences =
((YearOne_total_gross_monthly_income) * (rentalList.vacancy / 100));
var YearOne_property_taxes_expences =
(rentalList.property_taxes_expences * 12); var
YearOne_property_insurance_expences =
((rentalList.property_insurance_expences) * 12); var
YearOne_property_management_expences =
((YearOne_total_gross_monthly_income) *
(rentalList.property_management / 100));
//$scope.rentalAnnualYearOne.repairs_maintenance_expences =
Math.round((parameters.repairs_maintenance_expences) * 12); var
YearOne_repairs_maintenance_expences =
((YearOne_total_gross_monthly_income) *
(rentalList.repairs_maintenance / 100)); var YearOne_utilities =
(parseInt(rentalList.utilities) * 12); var YearOne_hoa =
(parseInt(rentalList.hoa) * 12); var YearOne_lawn_maintenance =
(parseInt(rentalList.lawn_maintenance) * 12); var
YearOne_snow_removal = (parseInt(rentalList.snow_removal) * 12);
var YearOne_expences = (rentalList.expences * 12); var
YearOne_total_operating_expenses = (YearOne_vacancy_expences +
YearOne_property_taxes_expences +
YearOne_property_insurance_expences +
YearOne_property_management_expences +
YearOne_repairs_maintenance_expences + YearOne_utilities +
YearOne_hoa + YearOne_lawn_maintenance + YearOne_snow_removal
+ YearOne_expences); var YearOne_net_operating_income =
(YearOne_total_gross_monthly_income -
YearOne_total_operating_expenses); var YearOne_annual_debt_service
= Math.round(((rentalList.annual_debt_service) * 12)); var
YearOne_cash_flow_before_taxes =
(parseInt(YearOne_net_operating_income) -
parseInt(YearOne_annual_debt_service)); /* End of the calculation */
/** Year five calculation **/ var rent_ncrease =
((rentalList.rent_ncrease / 100)); var rentPower = Math.pow((1 +
rent_ncrease), 4); var YearFive_monthly_rental_income =
(YearOne_monthly_rental_income * rentPower); var
YearFive_other_monthly_income = (YearOne_other_monthly_income *
```

```
rentPower); var YearFive_total_gross_monthly_income =
(YearFive_monthly_rental_income + YearFive_other_monthly_income);
/* Expences */ var expense_increase = ((rentalList.expense_increase /
100)); var expensePower = Math.pow((1 + expense_increase), 4);
//$scope.rentalAnnualYearFive.vacancy_expences =
Math.round(($scope.rentalAnnualYearOne.vacancy_expences *
expensePower)); var YearFive_vacancy_expences =
((YearFive_total_gross_monthly_income) * (rentalList.vacancy / 100));
var YearFive_property_taxes_expences =
Math.round((YearOne_property_taxes_expences * expensePower)); var
YearFive_property_insurance_expences =
Math.round((YearOne_property_insurance_expences * expensePower));
var YearFive_property_management_expences =
((YearFive_total_gross_monthly_income) *
(rentalList.property_management / 100)); var
YearFive_repairs_maintenance_expences =
((YearFive_total_gross_monthly_income) *
(rentalList.repairs_maintenance / 100)); var YearFive_utilities =
Math.round((YearOne_utilities * expensePower)); var YearFive_hoa =
Math.round((YearOne_hoa * expensePower)); var
YearFive_lawn_maintenance =
Math.round((YearOne_lawn_maintenance * expensePower)); var
YearFive_snow_removal = Math.round((YearOne_snow_removal *
expensePower)); var YearFive_expences =
Math.round((YearOne_expences * expensePower)); var
YearFive_total_operating_expenses = (YearFive_vacancy_expences +
YearFive_property_taxes_expences +
YearFive_property_insurance_expences +
YearFive_property_management_expences +
YearFive_repairs_maintenance_expences + YearFive_utilities +
YearFive_hoa + YearFive_lawn_maintenance + YearFive_snow_removal
+ YearFive_expences); var YearFive_net_operating_income =
(YearFive_total_gross_monthly_income -
YearFive_total_operating_expenses); var YearFive_annual_debt_service
= Math.round(((rentalList.annual_debt_service) * 12)); var
YearFive_cash_flow_before_taxes =
(parseInt(YearFive_net_operating_income) -
parseInt(YearFive_annual_debt_service)) /* End of the calculation */ /**
Year ten calculation **/ var rentPowerten = Math.pow((1 +
rent_ncrease), 9); var YearTen_monthly_rental_income =
(YearOne_monthly_rental_income * rentPowerten); var
YearTen_other_monthly_income = (YearOne_other_monthly_income *
rentPowerten); var YearTen_total_gross_monthly_income =
(YearTen_monthly_rental_income + YearTen_other_monthly_income); /*
```

```
Expences */ var expense_increase = ((rentalList.expense_increase /
100)); var expensePower = Math.pow((1 + expense_increase), 9);
//$scope.rentalAnnualYearTen.vacancy_expences =
Math.round(($scope.rentalAnnualYearOne.vacancy_expences *
expensePower)); var YearTen_vacancy_expences =
((YearTen_total_gross_monthly_income) * (rentalList.vacancy / 100));
var YearTen_property_taxes_expences =
Math.round((YearOne_property_taxes_expences * expensePower)); var
YearTen_property_insurance_expences =
Math.round((YearOne_property_insurance_expences * expensePower));
var YearTen_property_management_expences =
((YearTen_total_gross_monthly_income) *
(rentalList.property_management / 100));
//$scope.rentalAnnualYearTen.repairs_maintenance_expences =
Math.round(($scope.rentalAnnualYearOne.repairs_maintenance_expenc
* expensePower)); var YearTen_repairs_maintenance_expences =
((YearTen_total_gross_monthly_income) *
(rentalList.repairs_maintenance / 100)); var YearTen_utilities =
Math.round((YearOne_utilities * expensePower)); var YearTen_hoa =
Math.round((YearOne_hoa * expensePower)); var
YearTen_lawn_maintenance =
Math.round((YearOne_lawn_maintenance * expensePower)); var
YearTen_snow_removal = Math.round((YearOne_snow_removal *
expensePower)); var YearTen_expences =
Math.round((YearOne_expences * expensePower)); var
YearTen_total_operating_expenses = (YearTen_vacancy_expences +
YearTen_property_taxes_expences +
YearTen_property_insurance_expences +
YearTen_property_management_expences +
YearTen_repairs_maintenance_expences + YearTen_utilities +
YearTen_hoa + YearTen_lawn_maintenance + YearTen_snow_removal
+ YearTen_expences); var YearTen_net_operating_income =
(YearTen_total_gross_monthly_income -
YearTen_total_operating_expenses); var YearTen_annual_debt_service
= Math.round(((rentalList.annual_debt_service) * 12)); var
YearTen_cash_flow_before_taxes =
(parseInt(YearTen_net_operating_income) -
parseInt(YearTen_annual_debt_service)) /** End of the calculation **/
var date = new Date(); var newDate = date.getTime(); var filePath =
'./public/uploads/flip/' + newDate + '.pdf'; var options = {border: {
"top": "30px", // default is 0, units: mm, cm, in, px "right": "20px",
"bottom": "30px", "left": "20px" }}; var pdfFormData = '
'; pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
```

```
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
```

pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += ''; pdfFormData += '';
pdfFormData += ''; pdfFormData += '

REVENUE	**Monthly**
Monthly Rental Income	$' + commaNumber(rentalList.monthly_rental_income)
Other Income	$' + commaNumber(rentalList.other_monthly_income) +
Total Gross Monthly Income	$' + commaNumber(rentalList.total_gross_monthly_inco
EXPENSES	
Vacancy	$' + commaNumber(rentalList.vacancy_expences) + '
Real Estate Taxes	$' + commaNumber(rentalList.property_taxes_expences
Property Insurance	$' + commaNumber(rentalList.property_insurance_expe
Property Management	$' + commaNumber(rentalList.property_management_e
Repairs and Maintenance	$' + commaNumber(rentalList.repairs_maintenance_exp
Utilities	$' + commaNumber(rentalList.utilities) + '
HOA	$' + commaNumber(rentalList.hoa) + '
Lawn Maintenance	$' + commaNumber(rentalList.lawn_maintenance) + '
Snow Removal	$' + commaNumber(rentalList.snow_removal) + '
Other expenses	$' + commaNumber(rentalList.expences) + '
Total Operating Expenses	$' + commaNumber(rentalList.total_operating_expenses

Net Operating Income	'; pdfFormData += ''; pdfFormData += commaNumber(rentalList.net_operating_income.toStrin ')); pdfFormData += ''; pdfFormData += '
Less: Annual Debt Service	$' + commaNumber(rentalList.annual_debt_service) + '
Cash Flow Before Taxes	'; pdfFormData += ' '; pdfFormData += commaNumber(rentalList.cash_flow_before_taxes.toStri ', '')); pdfFormData += ' '; pdfFormData += '
Add Back: Principal Payment	N/A
Less: Depreciation	' + commaNumber(rentalList.depreciation) + '
Net Taxable Income	'; pdfFormData += ' '; pdfFormData += commaNumber(rentalList.net_taxable_income.toString(')); pdfFormData += ' '; pdfFormData += '
CAP Rate	'; pdfFormData += ' '; pdfFormData += '-'; pdfFormData += ' '; pdfFormData += '
Cash on Cash Return	'; pdfFormData += ' '; pdfFormData += '-'; pdfFormData += ' '; pdfFormData += '
Operating Expenses %	-
Cumulative Equity Buildup (Principal Paydown)	-
Cumulative Equity Buildup (Appreciation)	-
Market Value at End of Year	-
Outstanding Loan Balance	-

```
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'FINANCING CALCULATION'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Loan Amount'; pdfFormData += '
' pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.loan_amount);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Interest Rate (%)'; pdfFormData += '
' pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.interest_rate);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Loan Term (years)'; pdfFormData += '
' pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.loan_terms);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Monthly P&I Payment'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' +
commaNumber(rentalList.monthly_pi_payment); pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
```

'; pdfFormData += 'Monthly P&I Payment'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' +
commaNumber(rentalList.monthly_pi_payment); pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'CASH OUTLAY'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Purchase Price' pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.purchase_price);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Loan Amount'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.loan_amount);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Down Payment'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.down_payment_ex);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Repairs Costs'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.repair_costs);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Purchase Closing Fees'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' +

```
commaNumber(rentalList.purchase_closing_costs); pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Lender Fees'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.lender_fees)
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Lender Points'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.lender_points);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Appraisal Fee'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.appraisal_rate);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Home Inspection Fee'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' +
commaNumber(rentalList.home_inspection_fee); pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += 'Other Fee'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.other_fee);
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '

'; pdfFormData += 'Total Cash Outlay'; pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '$' + commaNumber(rentalList.total_cash_outlay);
```

```
pdfFormData += '
'; pdfFormData += '
'; pdfFormData += '
'; pdf.create(pdfFormData, options).toFile(filePath, function(err,
pdfinfo) { if (err) { console.log(err) } else { var transporter =
nodemailer.createTransport(config.development.connectionURL); var
baseUrl = req.protocol + '://' + req.get('host'); var options =
{template: req.body.content, replacement: {"{{user.name}}":
'Ashoishj', "{{logo_url}}": baseUrl + "/assets/images/black-logo.png",
"{{copyright}}": (new Date().getFullYear()) + " Copyright 2016", "
{{link.abuse_email}}": "info@resimpli.com"}}; var emailTemplate =
replace(options.template.toString(), options.replacement); var
message = { from: 'info@resimpli.com', to: req.body.to, cc:
req.body.cc, subject: req.body.subject, html: emailTemplate,
attachments: [ { filename: newDate + '.pdf', contentType:
'application/pdf', path: filePath } ] }; transporter.sendMail(message,
function(error, info) { if (error) { return console.log(error); }
res.json({status: constantObj.statusCode.success, message:
constantObj.messages.sendEmail}); }); } }); } else { return
res.send({status: constantObj.statusCode.error, message:
constantObj.messages.emptyCalculation}); } } }); } catch (err) {
res.json({status: constantObj.statusCode.error, message:
constantObj.messages.emptyData}); } } /** * Save Flip functionality *
@param {type} req * @param {type} res * @returns {unresolved} */
function save_flip(req, res) { /* Initialize Counter */ var reSet = '0'; var
data = rh.mapPost(req); data.users_id = req.user._id; /* Itemized
Calculation if user check */ if (data.checked === 'true') {
data.repair_cost = parseInt(data.permits); data.repair_cost +=
parseInt(data.mold); data.repair_cost += parseInt(data.asbestos);
data.repair_cost += parseInt(data.termites); data.repair_cost +=
parseInt(data.pest_control); data.repair_cost += parseInt(data.demo);
data.repair_cost += parseInt(data.waterproofing); data.repair_cost
+= parseInt(data.foundation); data.repair_cost +=
parseInt(data.septic_system); data.repair_cost += parseInt(data.roof);
data.repair_cost += parseInt(data.soffit_fascia_gutters);
data.repair_cost += parseInt(data.siding); data.repair_cost +=
parseInt(data.exterior_painting); data.repair_cost +=
parseInt(data.decks_porches_steps); data.repair_cost +=
parseInt(data.masonary); data.repair_cost +=
parseInt(data.concrete_asphalt); data.repair_cost +=
parseInt(data.garage_door); data.repair_cost +=
parseInt(data.landscaping); data.repair_cost += parseInt(data.fence);
data.repair_cost += parseInt(data.swimming_pool); data.repair_cost
+= parseInt(data.other_exterior_repairs); data.repair_cost +=
```

```
parseInt(data.plumbing); data.repair_cost +=
parseInt(data.electrical); data.repair_cost += parseInt(data.hvac);
data.repair_cost += parseInt(data.framing); data.repair_cost +=
parseInt(data.insulation); data.repair_cost += parseInt(data.drywall);
data.repair_cost += parseInt(data.doors_and_trims); data.repair_cost
+= parseInt(data.windows); data.repair_cost +=
parseInt(data.interior_painting); data.repair_cost +=
parseInt(data.kitchen_cabinets); data.repair_cost +=
parseInt(data.bathroom_vanity); data.repair_cost +=
parseInt(data.flooring); data.repair_cost += parseInt(data.tiling);
data.repair_cost += parseInt(data.appliances); data.repair_cost +=
parseInt(data.other_interior_repairs); } else { data.checked = "false";
} /* Basic Calulation */ data.maximum_purchase_price =
parseInt(data.resale_price) - (parseInt(data.minimum_required_profit)
+ parseInt(data.repair_cost)); /* Purchase Closing Calculation */ if
(data.purchase_closing_costs) data.purchase_costs =
(parseInt(data.purchase_closing_costs));
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
parseInt(data.purchase_closing_costs)); if (data.appraisal_rate)
data.purchase_costs = (parseInt(data.purchase_costs) +
parseInt(data.appraisal_rate)); data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
parseInt(data.appraisal_rate)); if (data.home_inspection_fee)
data.purchase_costs = (parseInt(data.purchase_costs) +
parseInt(data.home_inspection_fee)); data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
parseInt(data.home_inspection_fee)); if (data.other_fee)
data.purchase_costs = (parseInt(data.purchase_costs) +
parseInt(data.other_fee)); data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - parseInt(data.other_fee));
/* Loan Amount */ if (data.purchase_type === "Financing") { if
(data.loan_amount) data.loan_amount = parseInt(data.loan_amount); if
(data.interest_rate) data.interest_rate = data.interest_rate; if
(data.project_duration) data.project_duration =
parseInt(data.project_duration); if (data.lender_point)
data.lender_point = data.lender_point; /* Loan Interest Calculation */
//data.loan_interest = Math.round((parseInt(data.resale_price) *
parseInt(data.loan_amount) / 100 * parseInt(data.interest_rate) / 100 *
parseInt(data.project_duration) / 12)); data.loan_interest =
Math.round((parseInt(data.loan_amount) * data.interest_rate / 100 *
(parseInt(data.project_duration) / 12))); /* Lender Point Calculation */
data.lender_points = Math.round((data.lender_point / 100 *
parseInt(data.loan_amount))); /* Maximum Purchase Calculation */
```

```
data.maximum_purchase_price =
Math.round((parseInt(data.maximum_purchase_price) -
parseInt(data.loan_interest) - parseInt(data.lender_points) -
parseInt(data.lender_fees))); /* Total Loan Amount Calculation */
data.total_loan_amount = Math.round(parseInt(data.loan_amount)); /*
Lender Costs */ data.lender_costs =
Math.round((parseInt(data.loan_interest) +
parseInt(data.lender_points) + parseInt(data.lender_fees))); /* Project
Monthly Loan Payment */ data.monthly_loan_payment =
Math.round((parseInt(data.total_loan_amount) * (data.interest_rate /
12) / 100)); } else { data.loan_interest = parseInt(reSet);
data.interest_rate = parseInt(reSet); data.lender_point =
parseInt(reSet); data.lender_points = parseInt(reSet); data.lender_fees
= parseInt(reSet); data.total_loan_amount = parseInt(reSet);
data.monthly_loan_payment = parseInt(reSet); data.lender_costs =
parseInt(reSet); } /* Property Taxes */ if (data.property_taxes)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
(parseInt(data.property_taxes) * parseInt(data.project_duration)));
data.holding_cost = (parseInt(data.property_taxes) *
parseInt(data.project_duration)); if (data.electricity)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (parseInt(data.electricity) *
parseInt(data.project_duration))); data.holding_cost =
(parseInt(data.holding_cost) + (parseInt(data.electricity) *
parseInt(data.project_duration))); if (data.gas)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (parseInt(data.gas) *
parseInt(data.project_duration))); data.holding_cost =
(parseInt(data.holding_cost) + (parseInt(data.gas) *
parseInt(data.project_duration))); if (data.water)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (parseInt(data.water) *
parseInt(data.project_duration))); data.holding_cost =
(parseInt(data.holding_cost) + (parseInt(data.water) *
parseInt(data.project_duration))); if (data.garbage)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (parseInt(data.garbage) *
parseInt(data.project_duration))); data.holding_cost =
(parseInt(data.holding_cost) + (parseInt(data.garbage) *
parseInt(data.project_duration))); if (data.hoa)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (parseInt(data.hoa) *
parseInt(data.project_duration))); data.holding_cost =
```

```
(parseInt(data.holding_cost) + (parseInt(data.hoa) *
parseInt(data.project_duration))); if (data.property_insurance)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
(parseInt(data.property_insurance) * parseInt(data.project_duration)));
data.holding_cost = (parseInt(data.holding_cost) +
(parseInt(data.property_insurance) * parseInt(data.project_duration)));
if (data.lawn_care) data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (parseInt(data.lawn_care) *
parseInt(data.project_duration))); data.holding_cost =
(parseInt(data.holding_cost) + (parseInt(data.lawn_care) *
parseInt(data.project_duration))); if (data.other_holding_costs)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
(parseInt(data.other_holding_costs) *
parseInt(data.project_duration))); data.holding_cost =
(parseInt(data.holding_cost) + (parseInt(data.other_holding_costs) *
parseInt(data.project_duration))); /* Selling Costs */ if
(data.commission) data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) - (data.commission *
parseInt(data.resale_price) / 100)); data.comm =
Math.round((data.commission * parseInt(data.resale_price) / 100)); if
(data.closing_costs) data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
(parseInt(data.closing_costs))); if (data.Closing_costs_credit_to_buyer)
data.maximum_purchase_price =
(parseInt(data.maximum_purchase_price) -
(parseInt(data.Closing_costs_credit_to_buyer))); data.selling_cost =
(parseInt(data.comm) + parseInt(data.closing_costs) +
parseInt(data.Closing_costs_credit_to_buyer));
data.maximum_purchase_price = (data.maximum_purchase_price); /*
Personal Funds Needed */ data.total_funds_needed =
(parseInt(data.maximum_purchase_price) + parseInt(data.repair_cost)
+ parseInt(data.purchase_closing_costs) +
parseInt(data.appraisal_rate) + parseInt(data.home_inspection_fee) +
parseInt(data.other_fee)); /* Out of Pocket Funds/(Excess from Loan) */
if (data.total_loan_amount) { data.out_of_pocket =
(parseInt(data.total_funds_needed) -
parseInt(data.total_loan_amount)); } else { data.out_of_pocket =
data.total_funds_needed; } /* Net Profite */ data.net_profite =
((parseInt(data.resale_price) -
parseInt(data.maximum_purchase_price) -
parseInt(data.purchase_costs) - parseInt(data.repair_cost) -
parseInt(data.holding_cost) - parseInt(data.lender_costs) -
```

```
parseInt(data.selling_cost))); /* Project ROI */ data.project_roi =
(parseInt(((parseInt(data.net_profite) /
parseInt(data.total_funds_needed)) * 100).toFixed())); /* Project IRR */
if (data.project_duration) { data.project_irr =
Math.round((parseInt(data.project_roi) * (12 /
parseInt(data.project_duration)))); } else { data.project_irr =
parseInt(reSet); } var date = new Date(); var newDate =
date.getTime(); var filePath = './public/uploads/flip/' + newDate +
'.pdf'; var options = {format: 'Letter'}; var pdfFormData = 'Hello';
pdf.create(pdfFormData, options).toFile(filePath, function(err, res) { if
(err) { console.log(err) } else { var transporter =
nodemailer.createTransport(config.development.connectionURL); var
message = { from: 'isshu1987@gmail.com', to:
'resimpli.com@gmail.com', //uncomment when site is live subject: 'Flip
Analyzer', html: "Hello,
This is sample Demo", attachments: [ { filename: newDate + '.pdf',
contentType: 'application/pdf', path: filePath } ] };
transporter.sendMail(message, function(error, info) { if (error) {
return console.log(error); } console.log('Message sent: ' +
info.response); }); } }); Flip.findOne({users_id: req.user._id,
properties_id: data.properties_id}) .exec(function(err, flipList) { if
(!flipList) { var flipObj = new Flip(data); flipObj.save(function(err, flip)
{ if (err) { return res.send({status: constantObj.statusCode.error,
message: err}); } else { return res.send({status:
constantObj.statusCode.success, data: flip}); } }) } else { var query =
{_id: data._id}; Flip.findOneAndUpdate(query, data, function(err, data)
{ if (err) { return res.send({status: constantObj.statusCode.error,
message: err}); } else { return res.send({status:
constantObj.statusCode.success}); } }); } }); } /** * Flip Out
functionality * @param {type} req * @param {type} res * @returns
{unresolved} */ function getFlipOutput(req, res) { var FlipProjection
= { created: false, is_deleted: false, __v: false };
Flip.findOne({users_id: req.user._id, properties_id: req.params.id},
FlipProjection) .exec(function(err, flipList) { if (err) { return
res.send({status: constantObj.statusCode.error, message:
i18n.__('AUTH_ERROR')}); } else { if (flipList) { return
res.send({status: constantObj.statusCode.success, data: flipList}); }
else { return res.send({status: constantObj.statusCode.error, message:
constantObj.messages.emptyCalculation}); } } }); } /** * Save Rental
functionality * @param {type} req * @param {type} res * @returns
{unresolved} */ function save_rental(req, res) { var reSet = '0'; var
data = rh.mapPost(req); data.users_id = req.user._id; /* Rental
Calculation */ /* Itemized Calculation if user check */ if (data.checked
=== 'true') { data.repair_costs = parseInt(data.permits);
```

```
data.repair_costs += parseInt(data.mold); data.repair_costs +=
parseInt(data.asbestos); data.repair_costs += parseInt(data.termites);
data.repair_costs += parseInt(data.pest_control); data.repair_costs
+= parseInt(data.demo); data.repair_costs +=
parseInt(data.waterproofing); data.repair_costs +=
parseInt(data.foundation); data.repair_costs +=
parseInt(data.septic_system); data.repair_costs +=
parseInt(data.roof); data.repair_costs +=
parseInt(data.soffit_fascia_gutters); data.repair_costs +=
parseInt(data.siding); data.repair_costs +=
parseInt(data.exterior_painting); data.repair_costs +=
parseInt(data.decks_porches_steps); data.repair_costs +=
parseInt(data.masonary); data.repair_costs +=
parseInt(data.concrete_asphalt); data.repair_costs +=
parseInt(data.garage_door); data.repair_costs +=
parseInt(data.landscaping); data.repair_costs += parseInt(data.fence);
data.repair_costs += parseInt(data.swimming_pool); data.repair_costs
+= parseInt(data.other_exterior_repairs); data.repair_costs +=
parseInt(data.plumbing); data.repair_costs +=
parseInt(data.electrical); data.repair_costs += parseInt(data.hvac);
data.repair_costs += parseInt(data.framing); data.repair_costs +=
parseInt(data.insulation); data.repair_costs += parseInt(data.drywall);
data.repair_costs += parseInt(data.doors_and_trims);
data.repair_costs += parseInt(data.windows); data.repair_costs +=
parseInt(data.interior_painting); data.repair_costs +=
parseInt(data.kitchen_cabinets); data.repair_costs +=
parseInt(data.bathroom_vanity); data.repair_costs +=
parseInt(data.flooring); data.repair_costs += parseInt(data.tiling);
data.repair_costs += parseInt(data.appliances); data.repair_costs +=
parseInt(data.other_interior_repairs); } else { data.checked = "false";
} /* Loan Amount */ if (data.purchase_type === "Financing") {
data.loan_amount = (parseInt(data.purchase_price) * (1 -
data.down_payment / 100)); //data.loan_amount =
Math.abs(data.purchase_price * (1 - data.down_payment)); /* Pmt(
interest_rate, number_payments, PV, FV, Type ) Calculation */ var
interest = data.interest_rate / 100, // Annual interest years =
parseInt(data.loan_terms), // Lifetime of loan (in years) present =
data.loan_amount, // Present value of loan future = 0, // Future value
of loan beginning = 0; // Calculated at start of each period var
payment = -pmt(interest / 12, // Annual interest into months years *
12, // Total months for life of loan present, future, beginning);
data.monthly_pi_payment = Math.abs((payment)); data.lender_points
= Math.round((parseInt(data.loan_amount) * data.lender_point / 100));
} else { data.loan_amount = parseInt(reSet);
```

```
data.monthly_pi_payment = parseInt(reSet); data.down_payment =
parseInt(reSet); data.lender_point = parseInt(reSet);
data.lender_points = parseInt(reSet); data.lender_fees =
parseInt(reSet); data.interest_rate = parseInt(reSet); data.loan_terms
= parseInt(reSet); } /* Cash Outlay Calculation */
data.down_payment_ex = Math.abs((parseInt(data.purchase_price) -
parseInt(data.loan_amount))); data.total_cash_outlay =
total_cash_outlay(parseInt(data.down_payment_ex),
parseInt(data.repair_costs), parseInt(data.purchase_closing_costs),
parseInt(data.lender_fees), parseInt(data.lender_points),
parseInt(data.appraisal_rate), parseInt(data.home_inspection_fee),
parseInt(data.other_fee)); /* Monthly Revenue Calculation */
data.revenue = "Monthly"; data.total_gross_monthly_income =
(parseInt(data.monthly_rental_income) +
parseInt(data.other_monthly_income)); /* Expenses Calculation */
data.vacancy_expences = (data.vacancy *
data.total_gross_monthly_income / 100); data.property_taxes_expences
= (data.property_taxes / 12); data.property_insurance_expences =
(data.property_insurance / 12); data.property_management_expences
= (data.property_management * data.total_gross_monthly_income /
100); data.repairs_maintenance_expences =
(data.repairs_maintenance * data.total_gross_monthly_income / 100);
data.utilities = (parseInt(data.electricity) + parseInt(data.gas) +
parseInt(data.heating_oil) + parseInt(data.water) +
parseInt(data.garbage)); var Exp = parseInt(data.other_expenses) /
12; data.expences = (parseInt(data.owner_other_expenses) + Exp);
data.total_operating_expenses = (parseInt(data.vacancy_expences) +
parseInt(data.property_taxes_expences) +
parseInt(data.property_insurance_expences) +
parseInt(data.property_management_expences) +
parseInt(data.repairs_maintenance_expences) +
parseInt(data.utilities) + parseInt(data.expences) + parseInt(data.hoa)
+ parseInt(data.lawn_maintenance) + parseInt(data.snow_removal));
/* Net Operating Income */ data.net_operating_income =
(data.total_gross_monthly_income - data.total_operating_expenses)
data.annual_debt_service = data.monthly_pi_payment;
data.cash_flow_before_taxes = (parseInt(data.net_operating_income) -
parseInt(data.annual_debt_service)); data.principal_payment = "N/A";
data.depreciation = Math.round(((parseInt(data.purchase_price) /
27.5) / 12) * (1 - data.land_value / 100)); data.net_taxable_income =
Math.round((data.cash_flow_before_taxes - data.depreciation)); /* Save
Rental Calculation */ Rental.findOne({users_id: req.user._id,
properties_id: data.properties_id}) .exec(function(err, rentalList) { if
(!rentalList) { var rentalObj = new Rental(data);
```

rentalObj.save(function(err, rental) { if (err) { return res.send({status: constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); } else { return res.send({status: constantObj.statusCode.success, data: rental}); } }) } else { var query = {_id: data._id}; Rental.findOneAndUpdate(query, data, function(err, data) { if (err) { return res.send({status: constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); } else { return res.send({status: constantObj.statusCode.success}); } }); } }); console.log('data.loan_amount', data.total_cash_outlay); console.log('payment', Math.round(payment)); } /* PMT Calculation */ function pmt(rate_per_period, number_of_payments, present_value, future_value, type) { if (rate_per_period != 0.0) { // Interest rate exists var q = Math.pow(1 + rate_per_period, number_of_payments); return -(rate_per_period * (future_value + (q * present_value))) / ((-1 + q) * (1 + rate_per_period * (type))); } else if (number_of_payments != 0.0) { // No interest rate, but number of payments exists return -(future_value + present_value) / number_of_payments; } return 0; } /* Cash Outlay Calculation */ function total_cash_outlay(down_payment, repair_cost, purchase_closing_costs, lender_fees, lender_points, appraisal_fee, home_inspection_fee, other_fee) { return (down_payment + repair_cost + purchase_closing_costs + lender_fees + lender_points + appraisal_fee + home_inspection_fee + other_fee); } /** * Rental Out functionality * @param {type} req * @param {type} res * @returns {unresolved} */ function getRentalOutput(req, res) { var RentalProjection = { created: false, is_deleted: false, __v: false }; Rental.findOne({users_id: req.user._id, properties_id: req.params.id}, RentalProjection) .exec(function(err, rentalList) { if (err) { return res.send({status: constantObj.statusCode.error, message: i18n.__('AUTH_ERROR')}); } else { if (rentalList) { return res.send({status: constantObj.statusCode.success, data: rentalList}); } else { return res.send({status: constantObj.statusCode.error, message: constantObj.messages.emptyCalculation}); } } }); }