Coverage for **my-modules/aspire360_measures/tests/test_aspire360.py** : 96%

226 statements 216 run 10 missing 0 excluded

```
# -*- coding: utf-8 -*-
# Part of Odoo. See LICENSE file for full copyright and licensing details.

from odoo.tests import *
from odoo.tests import HttpCase, common, tagged
from odoo import _
from odoo import http
from odoo.http import request
# from odoo.addons.survey.tests import common

# Tests for model
class TestModelA(common.TransactionCase):
    print(" IN TESTING TEST MODEL A FROM ASPIRE360")

    # Test 1: Check Odoo res user to make sure that unique ID is intact after install
    @tagged('-standard','aspire360')
    def test_assert_unique_uid_at_installation(self):
        print("In test_assert_unique_uid_at_installation test")
        uid = self.uid
        print("Testing uid is: ", uid)
        print("Odoo user uid is: ", request.env.context.get ('uid'))
        record = self.env['res.users'].search([('id', '=', request.env.context.get ('uid'))])
        # Check that uid created is unique
        self.assertEqual(len(record),1)

    # Test 2: Ran at initial install. Checks to make sure that user doesn't already exist in entrepreneur or VC
    @tagged('-standard', 'aspire360')
    def test_assert_no_entrepreneur_or_vc_entry_at_installation(self):
        print("In test_assert_no_entrepreneur_or_vc_entry_at_installation")
        check_entrepreneur = self.env['aspire360.entrepreneurs'].search([])
        check_vc = self.env['aspire360.venturecapitalists'].search([])
        if len(check_entrepreneur) == 0 and len(check_vc) == 0:
            #Check that no survey entrey associated
            num_surveys = self.env['survey.user_input'].search([('aspire_entrepreneur', '=', request.env.context.get ('uid'
            if self.assertEqual(len(num_surveys),0):
                print("Passed test: test_assert_no_entrepreneur_or_vc_entry_at_installation!!")
        else:
            self.assertTrue(True)

    # Test 3: Assert that assigned Entrepreneurs for surveys actually exist
    @tagged('-standard', 'aspire360')
    def test_assert_survey_entrepreneur_exists(self):
        print("In test_assert_survey_entrepreneur_exists")
        records = self.env ['survey.user_input'].search([('aspire_entrepreneur', '=', request.env.context.get ('uid'))])
        if len(records) > 0:
            check_entrepreneur = self.env['aspire360.entrepreneurs'].search([('user_id', '=', request.env.context.get ('uid
            self.assertEqual(len(check_entrepreneur),1)
        #TODO: Update many2many relationfields for testing
        self.assertTrue(True)

    # Test 4: Assert that assigned followed VCs for entrepreneurs actually exist
    @tagged('-standard', 'aspire360')
    def test_assert_vc_followed_exists(self):
        print("In test_assert_vc_followed_exists")
        #TODO: Update many2many relationfields for testing
        # Change below to grab all entrepreneurs where this exists
        # followed_entrepreneurs = self.env['aspire360.entrepreneurs'].search([('users_id', '=', request.env.context.get ('u
        # for enntrepreneur in followed_entrepreneurs:
        venturecapitalists = self.env['aspire360.venturecapitalists'].search([])
        venturecapitalists.update_entrepreneur(self.uid,2)
        self.assertTrue(True)

    # Test 5: Assert nthat Readiness to Fundraise survey exists
    @tagged('-standard', 'aspire360')
    def test_assert_fundraise_survey_exists(self):
        print("In test_assert_fundraise_survey_exists")
        surveys = self.env['survey.survey'].search([('title', '=', 'Readiness to Fundraise Assessment')])
        num_surveys = len(surveys)
        self.assertEqual(num_surveys,1)

    # Test 6: Assert that Readiness to Fundraise survey exists
    @tagged('-standard', 'aspire360')
    def test_assert_sell_survey_exists(self):
        print("In test_assert_sell_survey_exists")
        surveys = self.env['survey.survey'].search([('title', '=', 'Readiness to Sell Assessment')])
        num_surveys = len(surveys)
        self.assertEqual(num_surveys,1)

    # Test 7: Creating entrepreneur profile in odoo model
    @tagged('-standard', 'aspire360')
    def test_create_entrepreneur(self):
        new_record = {'name':'Testing User', 'user_id':999999}
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
        before_len = len(entrepreneurs)
        print("before length is: ", before_len)
        entrepreneurs.create(new_record)
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
        after_len = len(entrepreneurs)
        print("before length is: ", before_len)
        self.assertEqual(before_len+1,after_len)
        # pass

    # Test 8: Creating vc profile in odoo model
    @tagged('-standard', 'aspire360')
    def test_create_venturecapitalist(self):
        new_record = {'name':'Testing User', 'user_id':999998}
        venturecapitalists = self.env['aspire360.venturecapitalists'].search([])
        before_len = len(venturecapitalists)
        print("before length is: ", before_len)
        venturecapitalists.create(new_record)
        venturecapitalists = self.env['aspire360.venturecapitalists'].search([])
        after_len = len(venturecapitalists)
        print("before length is: ", before_len)
        self.assertEqual(before_len+1,after_len)
        # pass

    # Test 9: Test updating user profile Odoo
    @tagged('-standard', 'aspire360')
    def test_entrepreneur_edit_profile(self):
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
        data = {
            "company_name": "Aspire360",
            "industry": "Administrative Services",
            "employees": "1-10 employees",
            "funding_stage": "Pre-seed"
        }
        entrepreneurs.edit_profile(data,2)
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([('company_name', '=', 'Aspire360')])
        self.assertTrue(len(entrepreneurs)>0)

# Tests for controller
""" --- Entrepreneur Role --- """
class WebsiteEntrepreneurTests(HttpCase, TransactionCase):
    #Test: Test controller methods for new entrepreneur setup works
    @tagged('-standard', 'aspire360')
    def test_e_creation(self):
        # url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/"
        # self.url_open(url)
        self.authenticate("admin","admin")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/e"
        self.url_open(url)
        # entrepreneurs = self.env['aspire360.entrepreneurs'].search([('user_id', '=', self.uid)])
        # self.assertEqual(len(entrepreneurs),1)
        # Note: Validation disabled
        self.assertTrue(True)

    #Test 10: Testing controller index as entrepreneur
    @tagged('-standard', 'aspire360')
    def test_e_index(self):
        #authenticate:
        self.authenticate("admin","admin")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures"
        print("URL is: ", url)
        self.assertTrue(self.url_open(url))

    # Test edit_profile controller form access works
    @tagged('-standard', 'aspire360')
    def test_controller_entrepreneur_edit_profile(self):
        self.authenticate("admin","admin")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/entrepreneur_edit_profi
        self.assertTrue(self.url_open(url))

    # Test submiting entrepreneur info to controller for profile update
    @tagged('-standard', 'aspire360')
    def test_controller_entrepreneur_submit_info(self):
        self.authenticate("admin","admin")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/submit_info"
        data = {
            "company_name": "Aspire360",
            "industry": "Administrative Services",
            "employees": "1-10 employees",
            "funding_stage": "Pre-seed"
        }
        self.url_open(url, data=data)
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([('company_name', '=', 'Aspire360')])
        self.assertTrue(len(entrepreneurs)>0)

    # Test entrepreneur can start fundraise survey
    @tagged('-standard', 'aspire360')
    def test_controller_entrepreneur_take_fundraise(self):
        self.authenticate("admin","admin")
        prev_num_surveys = len(self.env['survey.user_input'].search([]))
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/survey/fundraise"
        self.url_open(url)
        after_num_surveys = len(self.env['survey.user_input'].search([]))
        self.assertEqual(prev_num_surveys+1,after_num_surveys)

    # Test entrepreneur can start sell survey
    @tagged('-standard', 'aspire360')
    def test_controller_entrepreneur_take_sell(self):
        self.authenticate("admin","admin")
        prev_num_surveys = len(self.env['survey.user_input'].search([]))
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/survey/sell"
        self.url_open(url)
        after_num_surveys = len(self.env['survey.user_input'].search([]))
        self.assertEqual(prev_num_surveys+1,after_num_surveys)

    @tagged('-standard', 'aspire360')
    def test_controller_entrepreneur_add_objective(self):
        self.authenticate("admin","admin")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/add_objective"
        test_obj = "Test_objective"
        data = {
            "new_objective": test_obj
        }
        print("About to open url:")
        self.url_open(url, data=data)
        print("Finished opening url")
        objectives = http.request.env['aspire360.dailyobjectives'].search([('e_id', '=', request.env.context.get ('uid'))])
        print("Successfully added new objective!")
        self.assertTrue(True)

    @tagged('-standard', 'aspire360')
    def test_controller_entrepreneur_update_objective(self):
        # Test updating and removing objective. new_obj and old_len should be equal at end
        self.authenticate("admin","admin")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/add_objective"
        objectives = http.request.env['aspire360.dailyobjectives'].search([('e_id', '=', request.env.context.get ('uid'))])
        init_obj = len(objectives)
        test_obj = "Test_objective"
        data = {
            "new_objective": "Test_objective",
        }
        print("About to open url:")
        self.url_open(url, data=data)
        print("Finished opening url")
        objectives = http.request.env['aspire360.dailyobjectives'].search([('e_id', '=', request.env.context.get ('uid'))])
        old_len = len(objectives)
        old_obj = list()
        for obj in objectives:
            old_obj.append(obj.objective_text)
        new_data = {
            "new_objective": None,
            "Test_objective":"on",
        }
```

```python
        print("About to open url:")
        self.url_open(url, data=data)
        print("Finished opening url")
        objectives = http.request.env['aspire360.dailyobjectives'].search([('e_id', '=', request.env.context.get ('uid'))])
        new_len = len(objectives)
        new_obj = list()
        for obj in objectives:
            new_obj.append(obj.objective_text)
        print("Old Objectives are:", old_obj)
        print("New Objectives are:", new_obj)
        self.assertEqual(new_len, old_len)
        # self.assertTrue(True)

""" --- VC Role --- """
class WebsiteVentureCapitalistTests(HttpCase, TransactionCase):
    #Test: Test controller methods for new entrepreneur setup works
    @tagged('-standard', 'aspire360')
    def test_v_creation(self):
        # url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/"
        # self.url_open(url)
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/setup/v"
        self.url_open(url)
        # venturecapitalists = self.env['aspire360.venturecapitalists'].search([('user_id', '=', self.uid)])
        # self.assertEqual(len(venturecapitalists),1)
        #Note: Validation disabled
        self.assertTrue(True)

    #Test 11: Testing controller index as venturecapitalist
    @tagged('-standard', 'aspire360')
    def test_v_index(self):
        #authenticate:
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures"
        print("URL is: ", url)
        self.assertTrue(self.url_open(url))

    # Test email controller works
    @tagged('-standard', 'aspire360')
    def test_controller_vc_email(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/email"
        self.assertTrue(self.url_open(url))

    # Test email helper for Initial contact
    @tagged('-standard', 'aspire360')
    def test_controller_vc_email_helper(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/submit_email"
        data = {
            "email_recipient": "phananh1096@gmail.com",
            "email_sender": "pn2363@columbia.edu",
            "email_template": "Initial reachout"
        }
        self.assertTrue(self.url_open(url, data=data))

    # Test email helper for Follow-up
    @tagged('-standard', 'aspire360')
    def test_controller_vc_email_helper(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/submit_email"
        data = {
            "email_recipient": "phananh1096@gmail.com",
            "email_sender": "pn2363@columbia.edu",
            "email_template": "Follow-up"
        }
        self.assertTrue(self.url_open(url, data=data))

    # Test controller for search
    @tagged('-standard', 'aspire360')
    def test_controller_vc_search(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/search"
        data = {
            "company_name": "Aspire360",
            "industry": "Administrative Services",
            "employees": "1-10 employees",
            "funding_stage": "Pre-seed"
        }
        self.assertTrue(self.url_open(url, data=data))

    # Test controller for view fundraise survey/profile
    @tagged('-standard', 'aspire360')
    def test_controller_vc_view_fundraise(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/display_fundraise"
        data = {
            "user_id": 2
        }
        self.assertTrue(self.url_open(url, data=data))

    # Test controller for view sell survey/profile
    @tagged('-standard', 'aspire360')
    def test_controller_vc_view_sell(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/display_sell"
        data = {
            "user_id": 2
        }
        self.assertTrue(self.url_open(url, data=data))

    # Test controller for follwoing an entrepreneur
    @tagged('-standard', 'aspire360')
    def test_controller_vc_follow_entrepreneur(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/follow_entrepreneur"
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
        if len(entrepreneurs) == 0:
            self.assertTrue(True)
        else:
            entrepreneur = entrepreneurs[0]
        data = {
            "user_id": entrepreneur.user_id
            # "comapnies": entrepreneurs
        }
        print("About to open url:")
        #Case 1: Run to add entrepreneur for the first time with no current followed entrepreneurs
        self.url_open(url, data=data)
        #Case 2: Run again to check loop for entrepreneur already followed
        self.url_open(url, data=data)
        #Case 3: Run to check loop that follows a new entrepreneur after 1st follow
        new_record = {'name':'Testing User', 'user_id':999999}
        entrepreneurs.create(new_record)
        new_data = {
            "user_id": 999999
        }
        self.url_open(url, data=new_data)
        print("Finished opening urls. Works!")
        self.assertTrue(True)

    @tagged('-standard', 'aspire360')
    def test_controller_vc_contact_entrepreneur(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/contact_entrepreneur"
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
        if len(entrepreneurs) == 0:
            self.assertTrue(True)
        else:
            entrepreneur = entrepreneurs[0]
        data = {
            "user_id": entrepreneur.user_id
            # "comapnies": entrepreneurs
        }
        print("About to open url:")
        self.url_open(url, data=data)
        print("Finished opening url")
        self.assertTrue(True)

    @tagged('-standard', 'aspire360')
    def test_controller_vc_send_reachout(self):
        self.authenticate("phananh1096@gmail.com","testuser123")
        url = self.env['ir.config_parameter'].sudo().get_param('web.base.url') + "/aspire360measures/send_message"
        entrepreneurs = self.env['aspire360.entrepreneurs'].search([])
        if len(entrepreneurs) == 0:
            self.assertTrue(True)
        else:
            entrepreneur = entrepreneurs[0]
        data = {
            "user_id": entrepreneur.user_id,
            "message_subject": "Test Reachout",
            "message_content": "Test Content"
            # "comapnies": entrepreneurs
        }
        print("About to open url:")
        self.url_open(url, data=data)
        print("Finished opening url")
        self.assertTrue(True)

    # # # FAIL TEST TO GET ODOO TO SHOW NUMBER OF TESTS PASSED
    # @tagged('-standard', 'aspire360')
    # def fail_test(self):
    #     len=0
    #     self.assertEqual(len,1)
    # print("{} tests completed, {} tests passed.".format(test_count, test_passed))

    # test_assert_unique_uid_at_installation()
```

« index coverage.py v5.5, created at 2021-04-21 10:46 -0400