TEST CASES

class TestUrls(SimpleTestCase):  
  
 def test\_home\_url\_is\_resolved(self):  
 url = reverse('home')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, home)  
  
 def test\_signup\_url\_is\_resolved(self):  
 url = reverse('signup')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, signup)  
  
 def test\_signin\_url\_is\_resolved(self):  
 url = reverse('signin')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, signin)  
  
 def test\_signout\_url\_is\_resolved(self):  
 url = reverse('signout')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, signout)  
  
 def test\_openings\_url\_is\_resolved(self):  
 url = reverse('openings')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, opening)  
  
 def test\_signup\_url\_is\_resolved(self):  
 url = reverse('signup')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, signup)  
  
 def test\_candidates\_url\_is\_resolved(self):  
 url = reverse('candidates')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, candidates)  
  
 def test\_pipeline\_url\_is\_resolved(self):  
 url = reverse('pipeline')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, pipeline)  
  
 def test\_placement\_url\_is\_resolved(self):  
 url = reverse('placement')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, placement)  
  
 def test\_account\_url\_is\_resolved(self):  
 url = reverse('account')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, account)  
  
 def test\_dashboard\_url\_is\_resolved(self):  
 url = reverse('dashboard')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, dashboard)  
  
 def test\_addopenings\_url\_is\_resolved(self):  
 url = reverse('addopenings')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, addopenings)  
  
 def test\_addcandidates\_url\_is\_resolved(self):  
 url = reverse('addcandidates')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, addcandidates)  
  
 def test\_addpipeline\_url\_is\_resolved(self):  
 url = reverse('addpipeline')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, addpipeline)  
  
 def test\_addplacement\_url\_is\_resolved(self):  
 url = reverse('addplacement')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, addplacement)  
  
 def test\_addaccount\_url\_is\_resolved(self):  
 url = reverse('addaccount')  
 print(resolve(url))  
 self.assertEqual(resolve(url).func, addaccount)

class TestView(TestCase):  
  
 def setUp(self):  
 self.client = Client()  
 self.home\_url = reverse('home')  
 self.signup\_url = reverse('signup')  
 self.signin\_url = reverse('signin')  
 self.openings\_url = reverse('openings')  
 self.candidates\_url = reverse('candidates')  
 self.pipeline\_url = reverse('pipeline')  
 self.placement\_url = reverse('placement')  
 self.account\_url = reverse('account')  
  
  
 def test\_home(self):  
 response = self.client.get(self.home\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/index.html')  
  
 def test\_signup(self):  
 response = self.client.get(self.signup\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/signup.html')  
  
 def test\_signin(self):  
 response = self.client.get(self.signin\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/signin.html')  
  
 def test\_openings(self):  
 response = self.client.get(self.openings\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/openings.html')  
  
 def test\_candidates(self):  
 response = self.client.get(self.candidates\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/candidates.html')  
  
 def test\_pipeline(self):  
 response = self.client.get(self.pipeline\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/pipeline.html')  
  
 def test\_placement(self):  
 response = self.client.get(self.placement\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/placement.html')  
  
 def test\_account(self):  
 response = self.client.get(self.account\_url)  
  
 self.assertEqual(response.status\_code, 200)  
 self.assertTemplateUsed(response, 'ApplicantTracking/account.html')