**Web-based Database \_ Part II**

**Module 1:**

I created a Django Project called “**part3**” in my CSX account (rmarzba) and then followed the provided instructions to set it up and create an App called “**RM**”. I updated the settings of projects according to the instructions.

**Module 2:**

I edited the **models.py** in my RM application, in order to create my 9 tables from Part I. The edited models.py is attached:

from django.db import models

from django.core.validators import MaxValueValidator, MinValueValidator,MinLengthValidator,MaxLengthValidator

class Department(models.Model):

DepartmentNo = models.IntegerField(primary\_key=True, unique= True, null=False)

DepartmentName\_Choices=(

('Marketing', 'Marketing'),

('HR Management', 'HR Management'),

('Production/Engineering', 'Production/Engineering'))

DepartmentName = models.CharField(

max\_length=50,

choices=DepartmentName\_Choices,

default="Marketing"

)

class Candidate(models.Model):

CandidateID = models.AutoField(primary\_key=True, unique= True, null=False)

Name= models.CharField(max\_length=100)

Major= models.CharField(max\_length=50)

LastDegree\_choices= (

('Ph.D.', 'Ph.D.'),

('Master', 'Master'),

('BS', 'BS'))

LastDegree = models.CharField(

max\_length=20,

choices=LastDegree\_choices,

default="BS"

)

# minimum for GPA is 0.00 and maximum is 4.00

GPA= models.FloatField(validators=[

MinValueValidator(0.00),

MaxValueValidator(4.00),

])

DateofBirth = models.DateField(auto\_now=False, auto\_now\_add=False)

Sex\_choices= (

('Male', 'Male'),

('Female', 'Female'))

Sex = models.CharField(

max\_length=10,

choices=Sex\_choices,

default="Male"

)

YearsExperirence = models.IntegerField(default=0)

Skills= models.CharField(max\_length=200,default="none")

IsHired= models.BooleanField(default=False)

class CandidateSSN(models.Model):

SSNID = models.AutoField(primary\_key=True, unique= True, null=False)

#SSN should be exactly 9 digits

SSN= models.IntegerField()

CandidateID=models.ForeignKey('Candidate',on\_delete=models.CASCADE)

class AcceptedCandidate(models.Model):

AccCandidateID = models.AutoField(primary\_key=True, unique= True, null=False)

Assigned= models.BooleanField(default=False)

SSNID= models.ForeignKey('CandidateSSN',on\_delete=models.CASCADE)

CandidateID=models.ForeignKey('Candidate',on\_delete=models.CASCADE)

class Employee(models.Model):

EmployeeNumber = models.AutoField(primary\_key=True, unique= True, null=False)

Name= models.CharField(max\_length=100)

#if a Department is deleted, the Employees' DepartmentNo would be set to null

DepartmentNo= models.ForeignKey('Department',on\_delete=models.SET\_NULL,null=True)

HiringDate = models.DateField(auto\_now=True, auto\_now\_add=False)

DateofBirth = models.DateField(auto\_now=False, auto\_now\_add=False)

Sex\_choices= (

('Male', 'Male'),

('Female', 'Female'))

Sex = models.CharField(

max\_length=10,

choices=Sex\_choices,

default="Male"

)

AccCandidateID= models.ForeignKey('AcceptedCandidate',on\_delete=models.SET\_NULL,null=True)

#ManagerID is a recursive foreign Key.

ManagerID= models.ForeignKey('self',on\_delete=models.SET\_NULL,null=True)

def \_\_str\_\_(self):

return self.Name

class Salary(models.Model):

SalaryId = models.AutoField(primary\_key=True, unique= True, null=False)

EmployeeNumber = models.ForeignKey('Employee',on\_delete=models.CASCADE)

Month = models.IntegerField(validators=[

MinValueValidator(1),

MaxValueValidator(12),

])

Year = models.IntegerField(validators=[

MinValueValidator(1950),

MaxValueValidator(2100),

])

class SalaryUnit(models.Model):

SalaryUId = models.AutoField(primary\_key=True, unique= True, null=False)

SalaryId= models.ForeignKey('Salary',on\_delete=models.CASCADE)

IsItIncome= models.BooleanField(default=True)

class Income(models.Model):

SalaryUId= models.ForeignKey('SalaryUnit',on\_delete=models.CASCADE)

amount= models.IntegerField()

description= models.CharField(max\_length=100)

class Deduction(models.Model):

SalaryUId= models.ForeignKey('SalaryUnit',on\_delete=models.CASCADE)

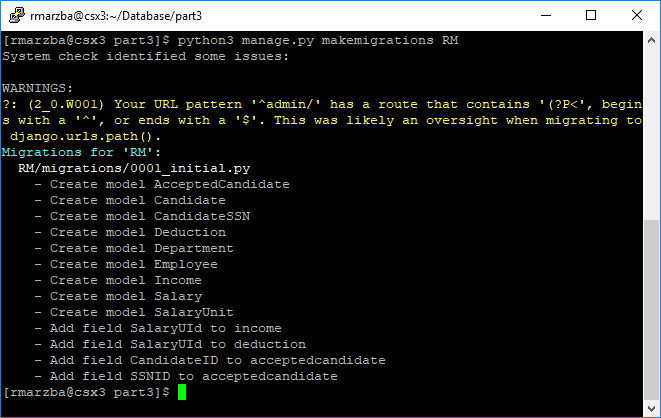
amount= models.IntegerField()

description= models.CharField(max\_length=100)

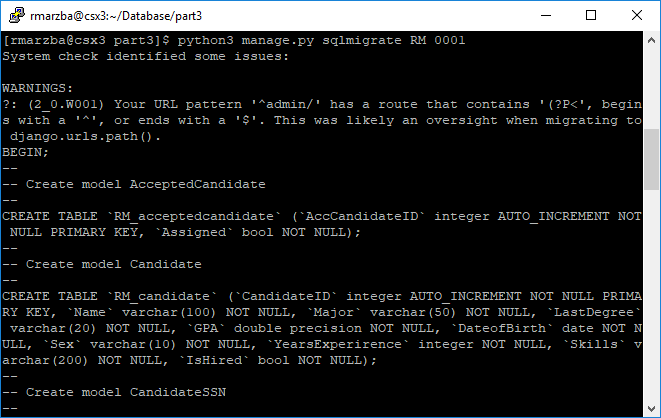
As you can see in our models.py, we have enforced the constraints like Domain, Range and Attribute constraints that I had mentioned in my Part I report.

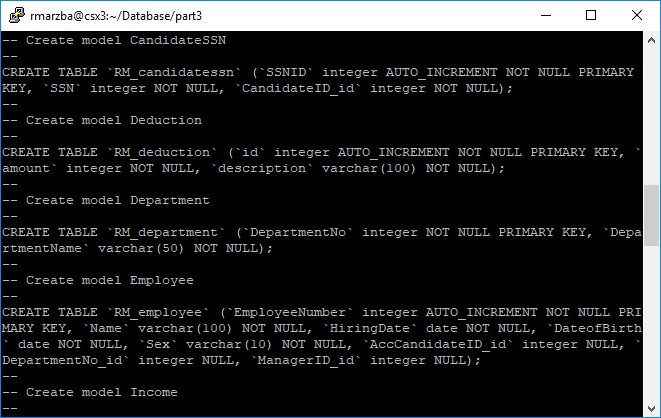
After saving the changes and migrating them, the SQL queries were created by Django. The screenshots are attached:

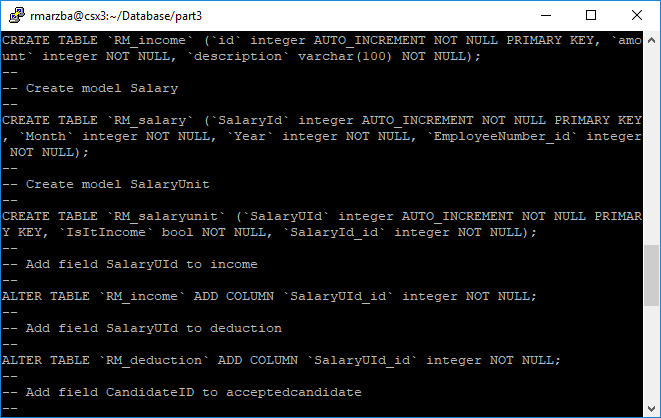
**Step1.** Makemigration from RM

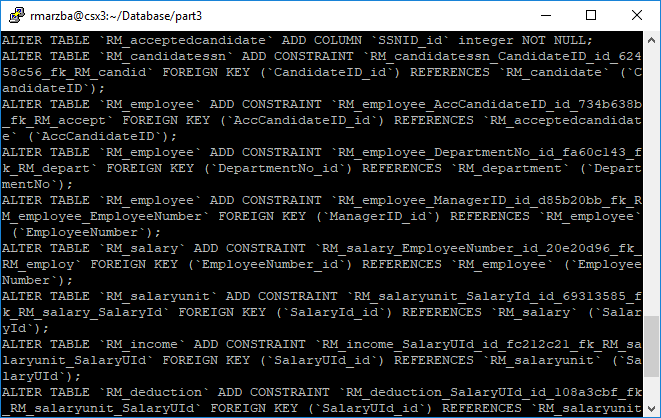


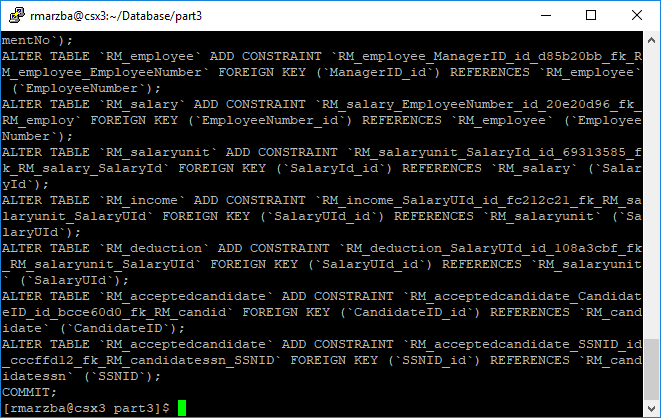
**Step2.** SQLmigrate from RM 0001 (the translated sql queries generated by Django.)



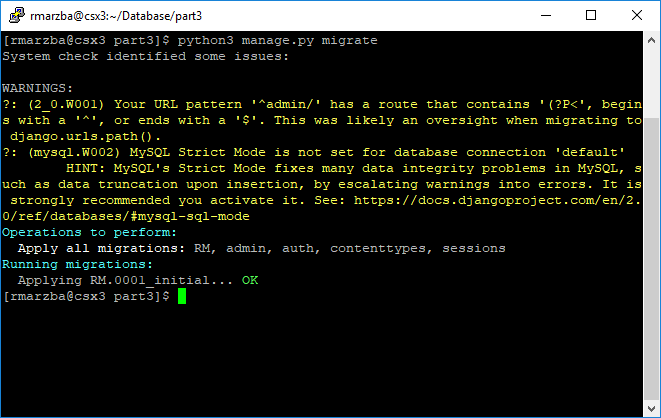




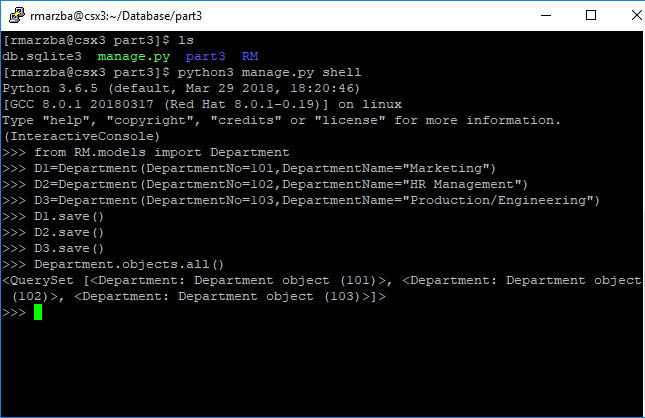




**Step3.** Migrate:



**Step4.** Entering Values for Department (Other tables would be filled with forms):



**Module 3:**

After creating and migrating the modules, I wrote the views according to them, the **views.py** final content is as follows:

from django.shortcuts import render, get\_object\_or\_404

from django.http import HttpResponse

from .forms import PayStubForm,CandidateEntryForm,CandidateSSNForm,IncomeForm,DeductionForm

from .models import Candidate,CandidateSSN,AcceptedCandidate,Employee,Department,Salary,SalaryUnit,Income,Deduction

from django.shortcuts import redirect

from django.utils import timezone

def base(request):

return render(request, 'base.html', {})

def insert\_candidate(request):

if request.method == "POST":

form = CandidateEntryForm(request.POST)

if form.is\_valid():

post = form.save(commit=False)

post.save()

return redirect('candidate\_ssn')

else:

form = CandidateEntryForm()

return render(request, 'newCandidate.html', {'form': form,})

def candidate\_ssn(request):

if request.method == "POST":

form = CandidateSSNForm(request.POST,request.FILES)

if form.is\_valid():

canID=Candidate.objects.filter(Name=form.cleaned\_data['Name'])

canID=canID[0]

candSSN=CandidateSSN(SSN=form.cleaned\_data['SSN'],CandidateID=canID)

candSSN.save()

return redirect('base')

else:

form= CandidateSSNForm()

return render(request, 'CandidateSSN.html', {'form': form,})

def candidate\_list(request):

candidates = Candidate.objects.all()

if request.method == "POST":

body = str(request.body)

body\_unicode=body.split('ID',1)[1]

body\_unicode= body\_unicode.replace("=", "")

ID= body\_unicode.replace("'", "")

ID=int(ID)

if "DELETE" in body:

Candidate.objects.filter(CandidateID=ID).delete()

else:

SSN=CandidateSSN.objects.filter(CandidateID=ID)

SSN=SSN[0]

Can=Candidate.objects.filter(CandidateID=ID)

Can=Can[0]

A=AcceptedCandidate(SSNID=SSN,CandidateID=Can)

A.save()

Candidate.objects.filter(CandidateID=ID).update(IsHired=True)

return redirect('candidate\_list')

return render(request, 'candidatelist.html', {'candidates': candidates,})

def ACCcandidate\_list(request):

Acandidates = AcceptedCandidate.objects.filter(Assigned=False)

if request.method == "POST":

body = str(request.body)

deptNo=int(body[-4:-1])

body=body.split('Department',1)[0]

body=body.split('ID',1)[1]

body= body.replace("=", "")

body= body.replace("'", "")

body= body.replace("&", "")

body=body.split('MID',1)

Can=int(body[0])

Mid=int(body[1])

Acandid=AcceptedCandidate.objects.filter(AccCandidateID=Can)[0]

candid=Acandid.CandidateID

Dept=Department.objects.filter(DepartmentNo=deptNo)[0]

if(Mid<=0): E=Employee(Name=candid.Name,DepartmentNo=Dept,HiringDate=timezone.now().date(),DateofBirth=candid.DateofBirth,

Sex=candid.Sex,AccCandidateID=Acandid)

else:

Manager=Employee.objects.filter(EmployeeNumber=Mid)[0] E=Employee(Name=candid.Name,DepartmentNo=Dept,HiringDate=timezone.now().date(),DateofBirth=candid.DateofBirth,

Sex=candid.Sex,AccCandidateID=Acandid,ManagerID=Manager)

E.save()

AcceptedCandidate.objects.filter(AccCandidateID=Can).update(Assigned=True)

return redirect('ACCcandidate\_list')

return render(request, 'AccCandidatelist.html', {'candidates': Acandidates,})

def employee\_list(request):

employees = Employee.objects.all()

return render(request, 'employeelist.html', {'employees': employees ,})

def PayStub(request):

form = PayStubForm()

return render(request, 'newPaystub.html', {'form': form,})

def PayStubView(request):

if(request.method == 'GET' and 'EmployeeNumber' in str(request.GET)):

get= str(request.GET)

body=get.split('EmployeeNumber',1)[1]

body=body.replace(":", "").replace("'", "").replace(",", "").replace(" ", "")

body=body.replace("[", "").replace("]", "").replace("}", "").replace(">", "")

body=body.split('Month',1)

ID=body[0]

body=body[1]

body=body.split('Year',1)

month=body[0]

year=body[1]

emp=Employee.objects.filter(EmployeeNumber=ID)[0]

if Salary.objects.filter(Year=year,Month=month,EmployeeNumber=emp).exists():

s1=Salary.objects.filter(Year=year,Month=month,EmployeeNumber=emp)[0]

else:

s1=Salary(Year=year,Month=month,EmployeeNumber=emp)

s1.save()

(IncomeT,DeductionT,incomes,deductions,T)=Get\_units(s1)

return render(request, 'Paystub.html', {'s1': s1,'T':T,'IncomeT':IncomeT,'DeductionT':DeductionT,'incomes':incomes,'deductions':deductions,})

def Add\_income(request):

if request.method=='GET':

s1= int(request.GET.get('s1'))

sa=Salary.objects.filter(SalaryId=s1)[0]

u=SalaryUnit(SalaryId=sa,IsItIncome=True)

u.save()

form = IncomeForm()

return render(request, 'incomeform.html', {'form': form,'s1': s1,'u': u.SalaryUId})

else:

return redirect('base')

def Add\_deduction(request):

if request.method=='GET':

s1= int(request.GET.get('s1'))

sa=Salary.objects.filter(SalaryId=s1)[0]

u=SalaryUnit(SalaryId=sa,IsItIncome=False)

u.save()

form = DeductionForm()

return render(request, 'deductionform.html', {'form': form,'s1': s1,'u': u.SalaryUId})

else:

return redirect('base')

def Add\_income2(request):

if(request.method == 'GET'):

s1= int(request.GET.get('s1'))

u= int(request.GET.get('u'))

amount= int(request.GET.get('amount'))

description= request.GET.get('description')

s1=Salary.objects.filter(SalaryId=s1)[0]

u=SalaryUnit.objects.filter(SalaryUId=u)[0]

a=Income(SalaryUId=u,amount=amount,description=description)

a.save()

(IncomeT,DeductionT,incomes,deductions,T)=Get\_units(s1)

return render(request, 'Paystub.html', {'s1': s1,'T':T,'IncomeT':IncomeT,'DeductionT':DeductionT,'incomes':incomes,'deductions':deductions,})

else:

return redirect('base')

def Add\_deduction2(request):

if(request.method == 'GET'):

s1= int(request.GET.get('s1'))

u= int(request.GET.get('u'))

amount= int(request.GET.get('amount'))

description= request.GET.get('description')

s1=Salary.objects.filter(SalaryId=s1)[0]

u=SalaryUnit.objects.filter(SalaryUId=u)[0]

a=Deduction(SalaryUId=u,amount=amount,description=description)

print('here')

a.save()

(IncomeT,DeductionT,incomes,deductions,T)=Get\_units(s1)

return render(request, 'Paystub.html', {'s1': s1,'T':T,'IncomeT':IncomeT,'DeductionT':DeductionT,'incomes':incomes,'deductions':deductions,})

else:

return redirect('base')

def Get\_units(s1):

units=SalaryUnit.objects.filter(SalaryId=s1)

incomes=[]

deductions=[]

IncomeT=0

DeductionT=0

for unit in units:

if(unit.IsItIncome):

i=Income.objects.filter(SalaryUId=unit)[0]

IncomeT=IncomeT+i.amount

incomes.append(i)

else:

i=Deduction.objects.filter(SalaryUId=unit)[0]

DeductionT=DeductionT+i.amount

deductions.append(i)

T=IncomeT-DeductionT

return (IncomeT,DeductionT,incomes,deductions,T)

The final content of **urls.py** is as follows:

from django.contrib import admin

from django.urls import path

# import url

from django.conf.urls import url

# import views

from RM import views

urlpatterns = [

path(r'AddIncome2/', views.Add\_income2, name='Add\_income2'),

path(r'AddIncome/', views.Add\_income, name='Add\_income'),

path(r'AddDeduction/', views.Add\_deduction, name='Add\_deduction'),

path(r'AddDeduction2/', views.Add\_deduction2, name='Add\_deduction2'),

path('PayStubView', views.PayStubView, name='PayStubView'),

path('PayStub', views.PayStub, name='PayStub'),

path('employeeList', views.employee\_list, name='employee\_list'),

path('AccCandidateList', views.ACCcandidate\_list, name='ACCcandidate\_list'),

path('candidateList', views.candidate\_list, name='candidate\_list'),

path('insertCan', views.insert\_candidate, name='insert\_candidate'),

path('insertCan/SSN', views.candidate\_ssn, name='candidate\_ssn'),

path(r'^admin/', admin.site.urls),

path('', views.base, name='base'),

]

**Templates:**

There were 10 total templates.

**1- AccCandidatelist.html**

<html>

<head>

<style>

#footer {

bottom: 0px;

background-color: #4CAF50;

width: 100%;

text-align: center;

}

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

#customers {

font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#customers td, #customers th {

border: 1px solid #ddd;

padding: 8px;

}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #4CAF50;

color: white;

}

</style>

</head>

<body>

<br>

<h1>Accepted Candidate List:</h1>

<table id="customers">

<tr><th>Accepted Candidate ID</th><th>Name</th><th>Experiens (years)</th><th>Skills</th><th>SSN</th></tr>

{% for can in candidates%}

<tr><td>{{ can.AccCandidateID }}</td><td>{{ can.CandidateID.Name}}</td><td>{{ can.CandidateID.YearsExperirence}}</td><td>{{ can.CandidateID.Skills}}</td><td>{{ can.SSNID.SSN}}</td></tr>

{% endfor %}

</table>

<hr>

<h1>Assign to a Department:</h1>

<p>Enter the Candiate ID and a Department Number to assign.</p>

<form method="POST">{% csrf\_token %}

Accepted candidate ID:<br>

<input type="text" name="ID"><br>

Manager Employee ID (Enter 0 if the manager is not specified):<br>

<input type="text" name="MID"><br>

Department Number (101,102,103):<br>

<input type="text" name="Department"><br>

<button type="submit" class="save btn btn-default">Submit</button>

</form>

<br>

<center><br><a href="{% url 'base' %}"><button class="button">Home Page</button></a><br><br></center>

<br><div id="footer"><b>

&copy; All rights reserved. Reza Marzban

</b>

</div>

<body>

</html>

**2- base.html**

<!DOCTYPE html>

<html>

<head>

<style>

#footer {

bottom: 0px;

background-color: #4CAF50;

width: 100%;

text-align: center;

}

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

</style>

</head>

<body>

<center>

<h2><b>CS 5423</b> - Database</h2>

<h3>Web-Based Database - Part III </h3>

<h2>Reza Marzban</h2>

</center>

<h3>Hiring:</h3>

<a href="{% url 'insert\_candidate' %}" target="\_blank"><button class="button">Enter a Candidate</button></a>

<a href="{% url 'candidate\_list' %}" target="\_blank"><button class="button">Candidates List</button></a>

<a href="{% url 'ACCcandidate\_list' %}" target="\_blank"><button class="button">Accepted Candidates List</button></a>

<br><br><br>

<h3>Employees:</h3>

<a href="{% url 'employee\_list' %}" target="\_blank"><button class="button">Current Employees List</button></a>

<br><br><br>

<h3>Pay-Stub:</h3>

<a href="{% url 'PayStub' %}" target="\_blank"><button class="button">Add/Edit/View Paystub</button></a>

<br><br><br>

<p>

<hr>

<h3>Help:</h3>

<ul>

<li>In order to Accept a candidate, go to candidate list and enter the candidate number.</li>

<li>Accepted Employees can be assigned to a Department and a manager in Accepted Candidates list.</li>

<li>You can add, edit or view a pay stub.</li>

</ul>

</p>

<br><div id="footer"><b>

&copy; All rights reserved. Reza Marzban

</b>

</div>

</body>

</html>

**3- candidatelist.html**

<html>

<head>

<style>

#footer {

bottom: 0px;

background-color: #4CAF50;

width: 100%;

text-align: center;

}

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

#customers {

font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#customers td, #customers th {

border: 1px solid #ddd;

padding: 8px;

}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #4CAF50;

color: white;

}

</style>

</head>

<body>

<br>

<h1>Candidate List:</h1>

<table id="customers">

<tr><th>Candidate ID</th><th>Name</th><th>Major</th><th>Last Degree</th><th>GPA</th><th>Date of Birth</th><th>Sex</th><th>Experiens (years)</th><th>Skills</th><th>Hired?</th></tr>

{% for can in candidates%}

<tr><td>{{ can.CandidateID }}</td><td>{{ can.Name}}</td><td>{{ can.Major}}</td><td>{{ can.LastDegree}}</td><td>{{ can.GPA}}</td><td>{{ can.DateofBirth}}</td><td>{{ can.Sex}}</td><td>{{ can.YearsExperirence}}</td><td>{{ can.Skills}}</td><td>{{ can.IsHired}}</td></tr>

{% endfor %}

</table>

<hr>

<h1>Delete/Hire a Candidate:</h1>

<p>Enter the Candiate ID to hire them or delete them.</p>

<form method="POST">{% csrf\_token %}

<select name="choose">

<option value="HIRE">Hire</option>

<option value="DELETE">Delete</option>

</select>

Candidate ID:<br>

<input type="text" name="ID"><br>

<button type="submit" class="save btn btn-default">Submit</button>

</form>

<center><br><a href="{% url 'base' %}"><button class="button">Home Page</button></a><br><br></center>

<br><div id="footer"><b>

&copy; All rights reserved. Reza Marzban

</b>

</div>

<body>

</html>

**4- CandidateSSN.html**

<!DOCTYPE html>

<html>

<head>

<style>

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

\* {

box-sizing: border-box;

}

input[type=text], select, textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

resize: vertical;

}

label {

padding: 12px 12px 12px 0;

display: inline-block;

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

float: right;

}

input[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

.col-25 {

float: left;

width: 25%;

margin-top: 6px;

}

.col-75 {

float: left;

width: 75%;

margin-top: 6px;

}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

@media screen and (max-width: 600px) {

.col-25, .col-75, input[type=submit] {

width: 100%;

margin-top: 0;

}

}

</style>

</head>

<body>

<h1>New Candidate (SSN)</h1>

<p>Enter your Social Security Number and confirm your name!</p>

<form method="POST">{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" class="save btn btn-default">Save</button>

</form>

</body>

</html>

**5- deductionform.html**

<!DOCTYPE html>

<html>

<head>

<style>

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

\* {

box-sizing: border-box;

}

input[type=text], select, textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

resize: vertical;

}

label {

padding: 12px 12px 12px 0;

display: inline-block;

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

float: right;

}

input[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

.col-25 {

float: left;

width: 25%;

margin-top: 6px;

}

.col-75 {

float: left;

width: 75%;

margin-top: 6px;

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

@media screen and (max-width: 600px) {

.col-25, .col-75, input[type=submit] {

width: 100%;

margin-top: 0;

}

}

</style>

</head>

<body>

<h1>Add an deduction salary unit</h1>

<form method="Get" action="/AddDeduction2/">

{{ form.as\_p }}

<input type="hidden" id="s1" name="s1" value={{ s1}}>

<input type="hidden" id="u" name="u" value={{ u}}>

<button type="submit" class="save btn btn-default">Add</button>

</form>

</body>

</html>

**6- employeelist.html**

<html>

<head>

<style>

#footer {

bottom: 0px;

background-color: #4CAF50;

width: 100%;

text-align: center;

}

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

#customers {

font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#customers td, #customers th {

border: 1px solid #ddd;

padding: 8px;

}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #4CAF50;

color: white;

}

</style>

</head>

<body>

<br>

<h1>Employee List:</h1>

<table id="customers">

<tr><th>Employee Number</th><th>Name</th><th>Department Name</th><th>Hiring Date</th><th>Date of Birth</th><th>Sex</th><th>Manager</th></tr>

{% for emp in employees%}

<tr><td>{{ emp.EmployeeNumber}}</td><td>{{ emp.Name}}</td><td>{{ emp.DepartmentNo.DepartmentName}}</td><td>{{ emp.HiringDate}}</td><td>{{ emp.DateofBirth}}</td><td>{{ emp.Sex}}</td><td>{{ emp.ManagerID.Name}}</td></tr>

{% endfor %}

</table>

<hr>

<center><br><a href="{% url 'base' %}"><button class="button">Home Page</button></a><br><br></center>

<br><div id="footer"><b>

&copy; All rights reserved. Reza Marzban

</b>

</div>

<body>

</html>

**7- incomeform.html**

<!DOCTYPE html>

<html>

<head>

<style>

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

\* {

box-sizing: border-box;

}

input[type=text], select, textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

resize: vertical;

}

label {

padding: 12px 12px 12px 0;

display: inline-block;

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

float: right;

}

input[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

.col-25 {

float: left;

width: 25%;

margin-top: 6px;

}

.col-75 {

float: left;

width: 75%;

margin-top: 6px;

}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

@media screen and (max-width: 600px) {

.col-25, .col-75, input[type=submit] {

width: 100%;

margin-top: 0;

}

}

</style>

</head>

<body>

<h1>Add an income salary unit</h1>

<form method="Get" action="/AddIncome2/">

{{ form.as\_p }}

<input type="hidden" id="s1" name="s1" value={{ s1}}>

<input type="hidden" id="u" name="u" value={{ u}}>

<button type="submit" class="save btn btn-default">Add</button>

</form>

</body>

</html>

**8- newCandidate.html**

<!DOCTYPE html>

<html>

<head>

<style>

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

\* {

box-sizing: border-box;

}

input[type=text], select, textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

resize: vertical;

}

label {

padding: 12px 12px 12px 0;

display: inline-block;

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

float: right;

}

input[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

.col-25 {

float: left;

width: 25%;

margin-top: 6px;

}

.col-75 {

float: left;

width: 75%;

margin-top: 6px;

}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

@media screen and (max-width: 600px) {

.col-25, .col-75, input[type=submit] {

width: 100%;

margin-top: 0;

}

}

</style>

</head>

<body>

<h1>New Candidate</h1>

<form method="POST">{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" class="save btn btn-default">Next</button>

</form>

</body>

</html>

**9- newPaystub.html**

<!DOCTYPE html>

<html>

<head>

<style>

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

\* {

box-sizing: border-box;

}

input[type=text], select, textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

resize: vertical;

}

label {

padding: 12px 12px 12px 0;

display: inline-block;

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

float: right;

}

input[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

.col-25 {

float: left;

width: 25%;

margin-top: 6px;

}

.col-75 {

float: left;

width: 75%;

margin-top: 6px;

}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

@media screen and (max-width: 600px) {

.col-25, .col-75, input[type=submit] {

width: 100%;

margin-top: 0;

}

}

</style>

</head>

<body>

<h1>Paystub Add/Edit/view</h1>

<form method="Get" action="{% url 'PayStubView' %}">{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" class="save btn btn-default">Add/Edit/view</button>

</form>

</body>

</html>

**10- Paystub.html**

<html>

<head>

<style>

#footer {

bottom: 0px;

background-color: #4CAF50;

width: 100%;

text-align: center;

}

.button {

display: inline-block;

padding: 15px 25px;

font-size: 24px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #4CAF50;

border: none;

border-radius: 15px;

box-shadow: 0 9px #999;

}

.button:hover {background-color: #3e8e41}

.button:active {

background-color: #3e8e41;

box-shadow: 0 5px #666;

transform: translateY(4px);

}

#customers {

font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 49.9%;

float:left;

}

#customers td, #customers th {

border: 1px solid #ddd;

padding: 8px;

}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #4CAF50;

color: white;

}

</style>

</head><body><br>

<h1>PayStub</h1>

<h2>The monthly salary paystub </h2>

<h3>Paystub information:</h3>

<p>Employee number:<b> {{ s1.EmployeeNumber.EmployeeNumber}}</b> &nbsp; &nbsp; Employee Name:<b> {{ s1.EmployeeNumber.Name}}</b> &nbsp; &nbsp; &nbsp; Month:<b> {{ s1.Month }}</b> &nbsp; Year:<b> {{ s1.Year }}</b> .

</p>

<h3>Salary Units List:</h3>

<table id="customers">

<tr><th colspan="2">Income</th></tr>

<tr><th>Description</th><th>Amount</th></tr>

{% for i in incomes%}

<tr><td>{{ i.description }}</td><td style="text-align:right">{{ i.amount}}</td></tr>

{% endfor %}

<tr><td><h3>Income Total:</h3></td><td style="text-align:right"><h3>{{ IncomeT}}</h3></td></tr>

</table>

<table id="customers" style="float:right;">

<tr><th colspan="2">Deduction</th></tr>

<tr><th>Description</th><th>Amount</th></tr>

{% for i in deductions%}

<tr><td>{{ i.description }}</td><td style="text-align:right">{{ i.amount}}</td></tr>

{% endfor %}

<tr><td><h3>Deduction Total:</h3></td><td style="text-align:right"><h3>{{ DeductionT}}</h3></td></tr>

</table>

<table id="customers" style="width: 100%;">

<tr><td><h2 style="text-align:right">Total:</h2></td><td style="text-align:right"><h2>{{ T }}</h2></td></tr>

</table>

<center>

<hr>

<p>&nbsp</p>

<a href="/AddIncome/?s1={{ s1.SalaryId}}"><button class="button">Adding a Income Salary unit</button></a>

<a href="/AddDeduction/?s1={{ s1.SalaryId}}"><button class="button">Adding a Deduction Salary unit</button></a>

</center><br><center>

<br><a href="{% url 'base' %}"><button class="button">Home Page</button></a><br><br></center>

<br><div id="footer"><b>

&copy; All rights reserved. Reza Marzban

</b>

</div>

<body>

</html>

**Forms:**

In order to create HTML forms based on our Django models I created a forms.py file in my application and the content is as follows:

from django import forms

from .models import Candidate, CandidateSSN,Salary,SalaryUnit,Income,Deduction

class CandidateEntryForm(forms.ModelForm):

class Meta:

model = Candidate

fields = ('Name','Major','LastDegree','GPA','DateofBirth','Sex','YearsExperirence','Skills')

labels= {'LastDegree':'Last Degree','DateofBirth':'Date of Birth', 'YearsExperirence':'Years Experience'}

class CandidateSSNForm1(forms.ModelForm):

class Meta:

model = CandidateSSN

fields = ('SSN',)

labels= {'SSN':'Social Security Number',}

class CandidateSSNForm(CandidateSSNForm1):

Name= forms.CharField()

class Meta(CandidateSSNForm1.Meta):

fields = CandidateSSNForm1.Meta.fields + ('Name',)

labels= CandidateSSNForm1.Meta.labels.update({'Name':'Confirm the candidate name',})

class PayStubForm(forms.ModelForm):

class Meta:

model = Salary

fields = ('EmployeeNumber','Month','Year')

labels= {'EmployeeNumber':'Choose the employee',}

class IncomeForm(forms.ModelForm):

class Meta:

model = Income

fields = ('amount','description')

class DeductionForm(forms.ModelForm):

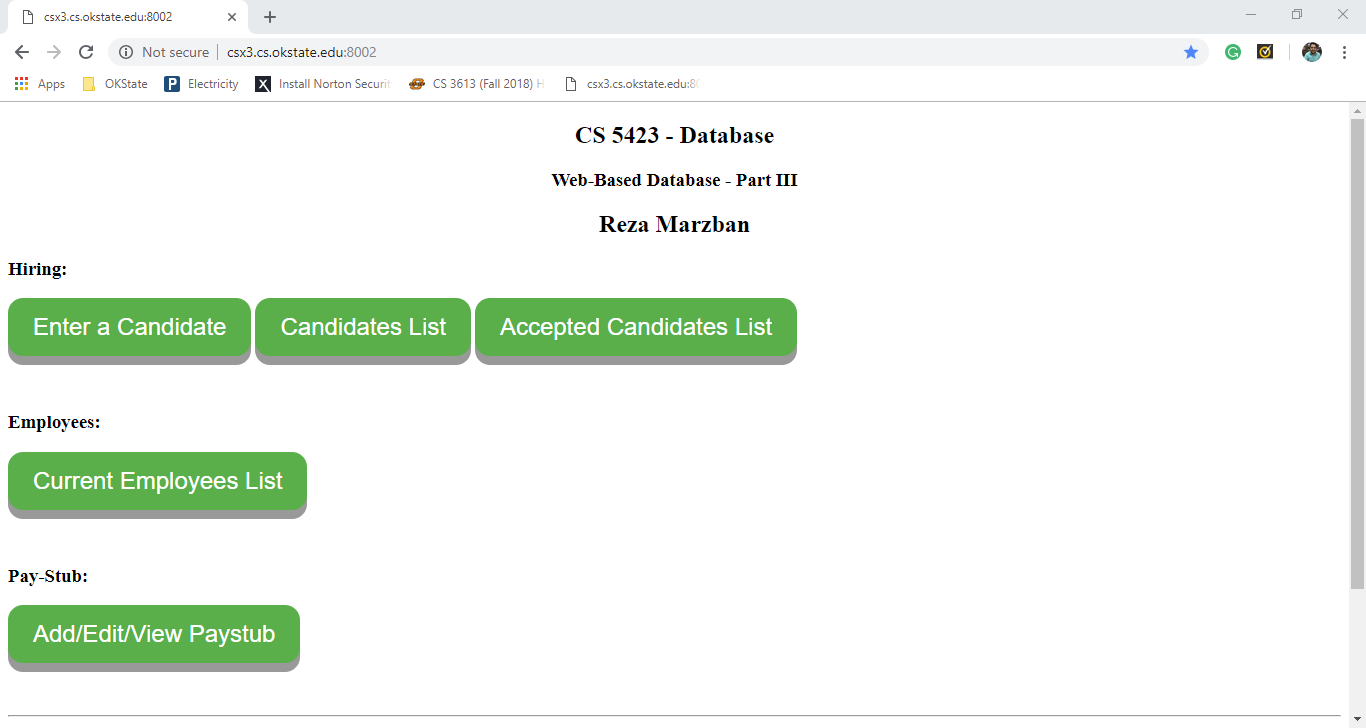
class Meta:

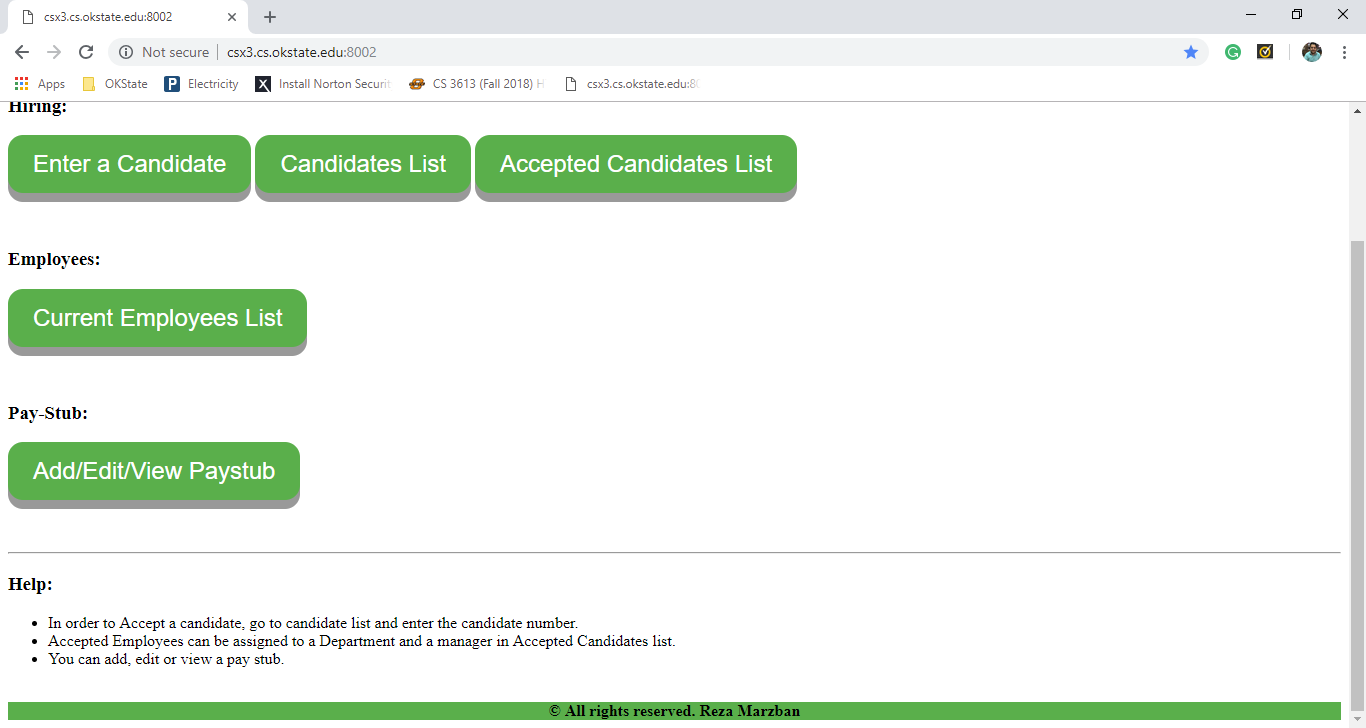
model = Deduction

fields = ('amount','description')

**Web-based program screenshots:**

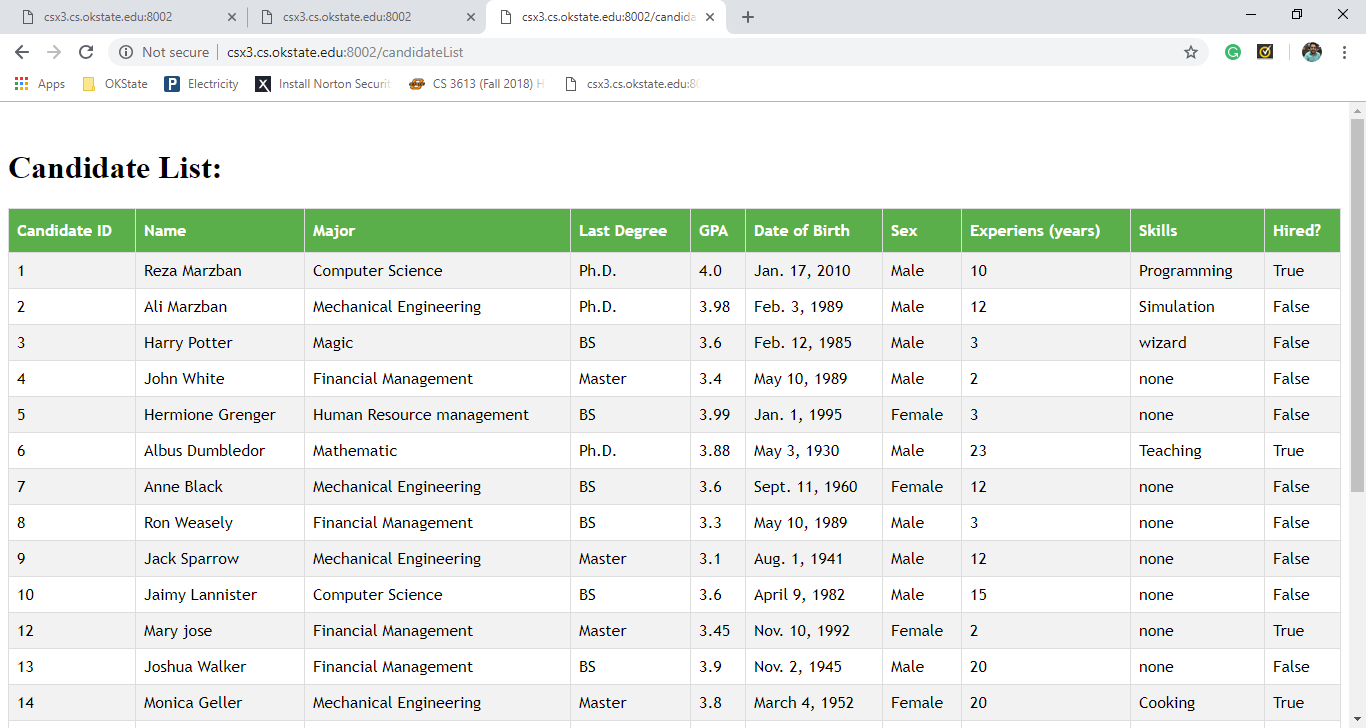
The program starts with a base page that acts as a homepage and the user can access different parts of program from there. And in every part of the program, there is a button the return the user to the base:

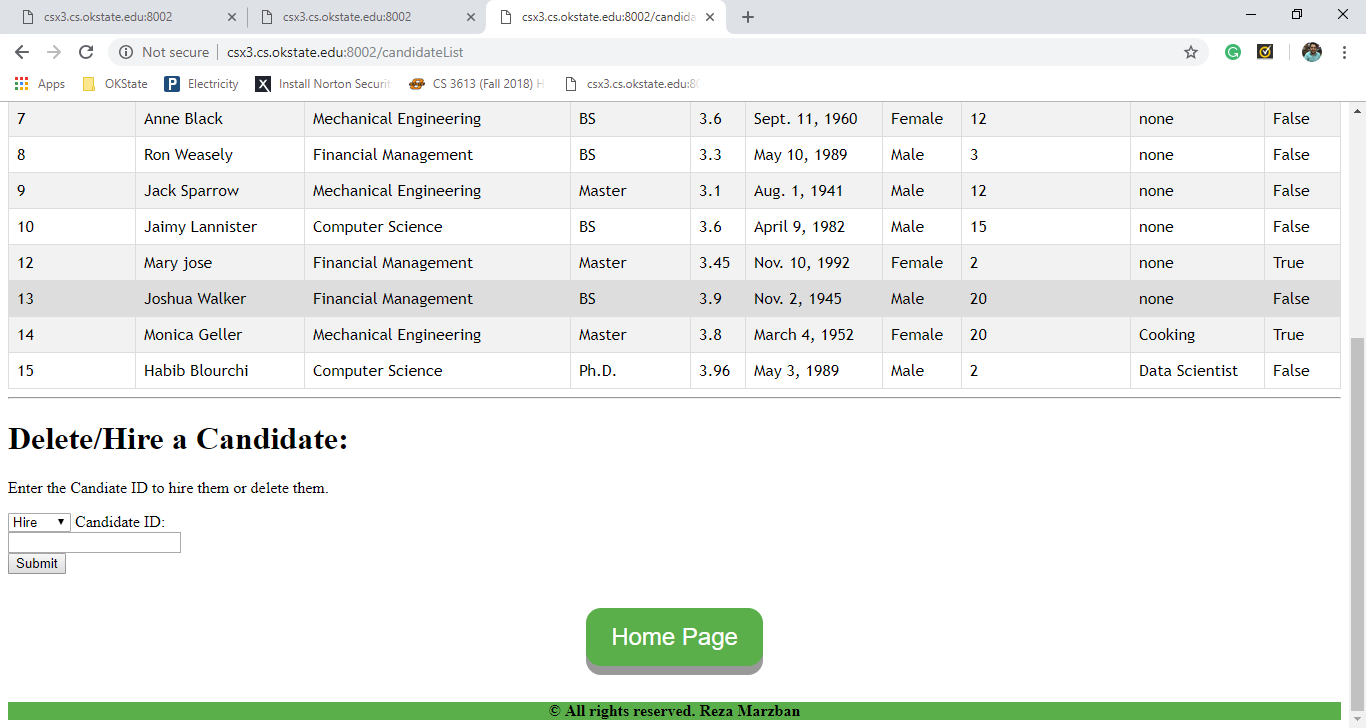




If you click on Enter a Candidate, there would be a 2-page form to insert a new candidate to two tables (candidate and candidateSSN).

If you click on Candidate List, you can see the list of all candidates, and in there you can choose to hire or delete a candidate:





In this page you can choose from drop down menu to hire or delete a candidate by giving their candidate ID.

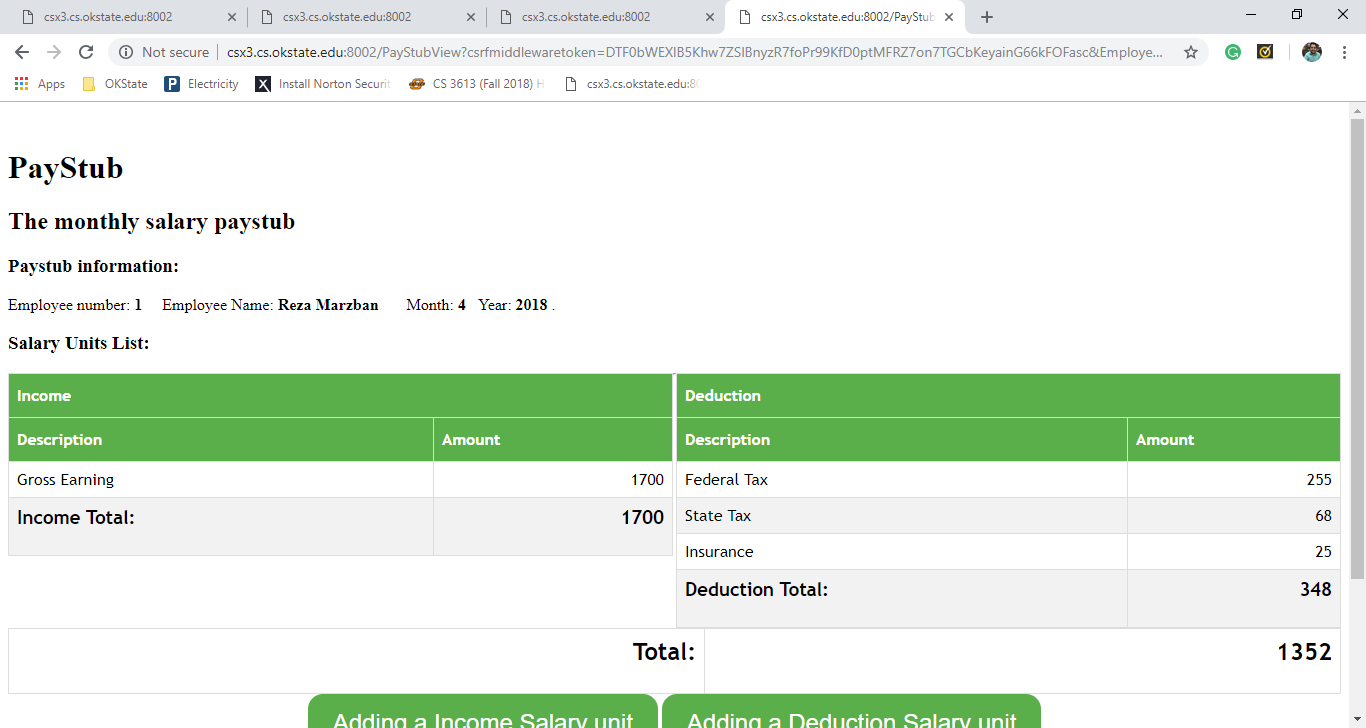
If you hire a candidate, they will be inserted into Acceptedcandidate that can be seen by clicking Accepted Candidate List from base and in there you can assign them to a department and a manager.

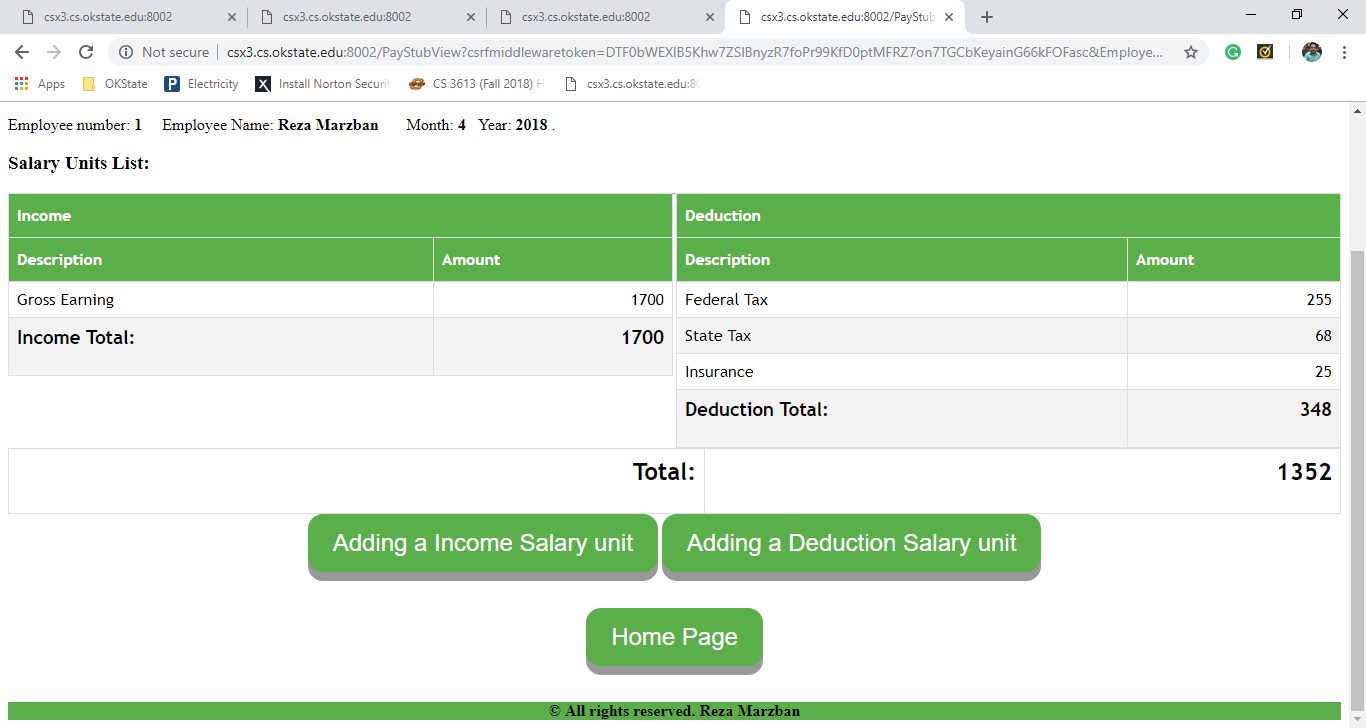
You can also see the Employee list from base.

The last part is Add/Edit/Show Pat stub, after clicking, the user has to choose the employee from drop down menu and also enter the year and month of salary to create or view/edit.

After opening a new Pay Stub, you can populate it with salary units that can be either Income or Deduction.

The Program automatically calculate the Total income, Total deductions and final Total.





The total is calculated by subtracting Deduction total from Income Total.