ECS 89

5/14

Announcements

- Thank you for entering pedometer data!

- Checkpoint did not go so great…
- Where did people get stuck?
- I'm going to just give you the code today
- Go over how to continue assignment from there

- Prof. Amenta extra office hour Mon 2-3
- Jesse's regular office hour Mon 4-5

Troubleshooting

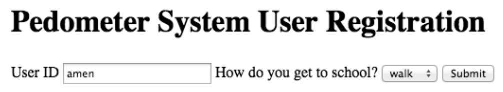
- This is a difficult environment
- Pages coming up different when you load them again → time to restart the server
- Pages not changing when you change the code → clear your browser cache
- 505 errors → something does not have permission 770; go to /var/www/yourname and do:
  chmod 770 –R mysite

Model for user

```
class User(models.Model):
   uid = models.CharField(max_length=20)
   transport = models.CharField(max_length=4,
         default="walk")
```

- What does the database corresponding to this look like?
- Goes in file models.py
- Remember to run "python manage.py syncdb"

User registration code


Pedometer System User Registration

User ID amen How do you get to school? walk Submit

- Form never changes; we don't need to pass any data in; so view can be pretty simple.

```
def uidForm(request):
   context = {'message': '' }
   return render(request, 'steps/uidForm.html', context)
```

All the work is in the template

```
<h1> Pedometer System User Registration </h1>

<strong> {{ message }} </strong>
<form action="gotForm" method="get">
…
<input type="submit" value="Submit">
</form>
```

Contents of the form

```
<label> User ID <input type="text" name="uid"> </label>
<label> How do you get to school? <select name="transport"">
  <option value="c">car</option>
  <option value="b">bus</option>
  <option value="k">bike</option>
  <option value="w">walk</option>
</select>
</label>
```

Page receiving data

- Lots of things can go wrong
- There might not be a uid and a transport
- The uid might already be in the database
- Use try-except-else to handle these

```
try:
      myid=request.GET['uid']
      trans=request.GET['transport']
   except:
      context = {"message": "Enter UID and make a selection"}
      return render(request, 'steps/uidForm.html', context)
```

Got the form data…now…

```
else:  # userid in myid, transport mode in trans
        try:
            u = User.objects.get(uid=myid)
        except: # there is no such user; add him or her
            u = User(uid=myid, transport=trans)
            u.save()
        else: # user is there already, change transport mode
            u.transport = trans
            u.save() # need for changes as well as new stuff!
```

Go directly to display page

```
context = {'userList': User.objects.all()}
return render(request, 'steps/uidAccept.html', context)
```

- Pass the whole list of users to a template and display it.
- Could have simply said “thank you”

Next steps

- We want to add in the pedometer data
- This is coming from steps.csv
- We can write a regular Python program to load objects into Django databases
- Put the program in /var/www/yourname

Load Django, settings, our classes

```
from django.core.management import setup_environ
from mysite import settings
setup_environ(settings)
from steps.models import User, Pedometer

# We can now write a normal python program that accesses
# our Django database.  For example:
#u = User(uid=”gump", transport=“k")
#u.save()
```

Class for pedometer data

```
class Pedometer(models.Model):
    user = models.ForeignKey(User)
    steps = models.PositiveIntegerField()
    month = models.PositiveIntegerField()
    day = models.PositiveIntegerField()
```

- The ForeignKey function indicates that this attribute is a relation to a row of the User table