# ECS 89

# Announcements

□ Checkpoint on Proj3 due Wednesday night (pushed back one day)

□ Set up user ID Django database, Web sites for user ID entry

# Pedometer data entry – use it!

**Enter your pedometer data:**



Steps: ____
User ID: ____
Month: ____
Day: ____
Submit

pc110.cs.ucdavis.edu:10002/hw2/index.html

# Last time

□ Django's Model class represents database table

□ We get data into the table by creating objects

□ Your assignment will involve filling in rows of user id table

□ Last time, we were getting data into and out of tables using commands in the interactive shell, but you should use the same commands in your code

□ These are all methods of Model

# Putting data into a table

□ Make a new choice object:

    c = Choice(choice_text="Yes", votes=0, poll=p)

    # p should be a poll object

□ Simultaneously, c is an object in your code, and there is a new row in the Choice table in the database. It'll remain there, on disk, until you delete it.

# Getting data out of tables

□ rs = Poll.objects.all() - gets everything

# returns an iterable object (list+extra stuff)

□ p = Poll.objects.get(id=1) – gets one particular item

# returns a single object

□ rs = Poll.objects.filter(question__contains("goal"))

    rs = Choice.objects.filter(choice_text="Yes")

## Now…building Web pages

□ First, tell Django your new app has urls.
□ In mysite/mysite/urls.py:

```
urlpatterns = patterns('',
    url(r'^polls/', include('polls.urls', namespace =
    'polls')),
    url(r'^newpolls/', include('newpolls.urls')),
    url(r'^admin/', include(admin.site.urls)),
)
```

## Add urls.py file to app

□ Add file mysite/newpolls/url.py:

```
from django.conf.urls import patterns, url
from newpolls import views
urlpatterns = patterns('',

    # matches /newpolls/detail/
    url(r'^detail/?$', views.detail, name='detail'),

    # matches /newpolls/vote/
    url(r'^vote/?$', views.vote, name='vote'),
)
```

## The url pattern specification

```
url(r'^detail/?$', views.detail, name='detail')
```

□ Begins with a regular expression that matches the url.
□ The url pattern we added in /mysite/mysite gets us to /django/newpolls/  (including the slash!)

## The url pattern specification

```
url(r'^detail/?$', views.detail, name='detail')
```

□ Regular expression means "has to start at the beginning, see string "detail", may or may not have a slash, then end"

## Views

□ These are the functions that get run when the HTTP request for that URL comes in
□ One job is to produce the output string that gets returned in the HTTP response, usually HTML (what else might it be?)
□ Might do something else as well, eg. put data into database

## Stub views

□ Add file /mysite/newsite/views.html:

```
from django.http import HttpResponse

def detail(request):
    return HttpResponse("This is where the poll will be.")

def vote(request):
    return HttpResponse("This is where you go after voting.")
```

## Templates

□ Allow you to put "variables" into HTML pages

□ Example – null.html:

```
<p>
{{ message }}
</p>
```

□ Say variable message = "Hello world".  Then this resulting Web page is:

```
<p>
   Hello world
</p>
```

## Template directory

□ Put all templates into directory mysite/newpolls/template/newpolls/

□ Django knows to look for them there

□ Use the Django render function to put variables into the template.

□ It takes three inputs: request (ignore for now), a template, and a dictionary giving the values of all the variables.

## Using the template in views.py

```python
from django.shortcuts import render
from newpolls.models import Poll, Choice

def details(request):
    p = Poll.objects.get(id=2)
    context = {'message': "The poll will go here" }
    return render(request, 'newpolls/null.html', context)

def votes(request):
    context = {'message': "You get here after voting."}
    return render(request, 'newpolls/null.html', context)
```