

Network Programming

- 網路程式設計 -

新聞公告系統 II

授課教師：張珀銀 老師



前一單元內容回顧

08

專題：新聞公告系統

8-1 簡易新聞公告系統

8-2 完整新聞公告系統



1

簡易新聞公告系統

8-1 簡易新聞公告系統

幾乎每一個網站都會使用新聞公告系統，可以將網站最新訊息快速告知所有瀏覽者。Django 內建完整新增、修改、刪除資料庫管理功能，簡易新聞公告系統可使用內建資料庫管理功能，就能輕鬆建立系統。

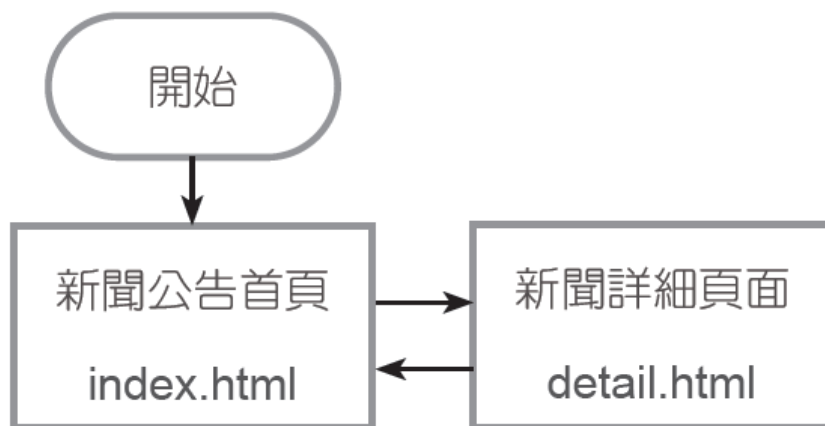
8.1.1 使用簡易新聞公告系統

如果資料庫沒有資料會顯示提示訊息：



8.1.2 簡易新聞公告系統流程圖

由於簡易新聞公告系統的管理介面使用Django 內建的管理功能，因此系統非常簡單，只有資料列表及詳細資料兩個頁面。



8.1.3 使用內建管理系統

簡易新聞公告系統的管理介面使用Django 內建的管理功能，使用者若要發布、編輯或刪除新聞內容，可以登入內建管理系統進行管理。本書專案管理者帳號皆為「admin」，密碼皆為「a123456!」。

請將本章範例news 專案複製到複製到<C:\example> 資料夾中，然後切換到news專案以「python manage.py runserver」啟動伺服器。

在瀏覽器輸入「http://127.0.0.1:8000/admin/」開啟登入頁面，輸入帳號及密碼後就可管理資料庫。本專案已預先建立18 筆資料，如果要發布新聞可點選**News units** 資料表右方的 **新增** 鈕：



內建管理系統設定檔

內建管理系統設定檔<admin.py> 預設不會顯示<models.py> 建立的資料表，必須在<admin.py> 註冊才會顯示。

```
newsapp\admin.py
```

```
1 from django.contrib import admin
2 from newsapp import models
3
4 admin.site.register(models.NewsUnit)
```


8.1.4 簡易新聞公告系統資料庫結構

資料庫結構定義在<models.py> 中，內含資料表NewsUnit：

```
newsapp\models.py
```

```
1 from django.db import models
2
3 class NewsUnit(models.Model): # 新聞資料表
4     catego = models.CharField(max_length=10, null=False) # 類別關聯
5     nickname = models.CharField(max_length=20, null=False) # 暱稱
6     title = models.CharField(max_length=50, null=False) # 標題
7     message = models.TextField(null=False) # 內容
8     pubtime = models.DateTimeField(auto_now=True) # 發布時間
9     enabled = models.BooleanField(default=False) # 是否顯示
10    press = models.IntegerField(default=0) # 點擊次數
11    def __str__(self):
12        return self.title
```

8.1.5 Url 配置檔

Url 配置檔<urls.py> 指定網址列執行的函式。簡易新聞公告系統只有新聞列表頁面(首頁) 及詳細頁面，<urls.py> 內容為：

news\urls.py

略……

```
16 from django.contrib import admin
17 from django.urls import path
18 from newsapp import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', views.index),    # 不帶參數的首頁 ( 預設第 1 頁 )
23     path('index/<str:pageindex>', views.index),    # 帶參數的首頁
24     path('detail/<int:detailid>', views.detail),    # 詳細頁面
25 ]
```

8.1.6 建立網頁基礎模版

網站中各網頁通常會具有相同的部分，例如同樣的頁首及頁尾，以維持網站一致性風格。如果在每一個網頁中都撰寫這些共同的部分，不但會大幅增加網頁程式碼的份量，更造成維護網站的負擔：若日後要修改網站風格，還必須修改每一個網頁的程式碼，基礎模版可以解決此問題。將各網頁相同的部分儲存為基礎模版，然後在各網頁中繼承基礎模版，未來若需要修改這些共同的部分，只要修改基礎模版即可。

簡易新聞公告系統的基礎模版為網頁上方的標題及下方的版權部分，將其儲存為 `<base.html>` 檔：



8.1.7 首頁處理函式及模版

使用者以「http://127.0.0.1:8000」執行本專案，或在首頁按 **上一頁**、**下一頁** 鈕，或在詳細頁面按 **首頁** 鈕，都會執行<views.py> 中的index 函式載入<index.html> 網頁。

8.1.8 詳細頁面處理函式及模版

使用者在首頁點選新聞標題就會執行<views.py> 中的detail 函式載入<detail.html> 網頁。



SPAN標記與DIV

HTML中的span標記和div標記

使用css排版網頁時，span和div常用的兩個的標記。利用這個兩個標記，加上css樣式控制

span和div概念區別和相似點

1. span標記與div標記一樣，作為容器標記廣泛應用在HTML中。span與中間同樣可以容納各種html元素，span與中間也可以視為獨立的對象。
2. span和div兩個標記都可以獨立出區塊，這一點沒有很大的區別。

SPAN標記與DIV

span和div功能區別

1. span和div區別在於，div是一個塊級元素，它包含的元素會自動換行。而span是行內元素，在它的前後不會換行。
2. span沒有結構上的意義，只是單純的應用樣式，其他元素不適合時，就可以使用span元素。
3. span可以作為div的子元素，但div不能是span的子元素，如果出現span中出現div不符合ws3c的頁面標準。

Result

← → ↻ 🏠 ⓘ localhost:63342/DJ_0413/templates/

DIV標記，圖片顯示分行



span標記，圖片顯示在同一行





2

完整新聞公告系統

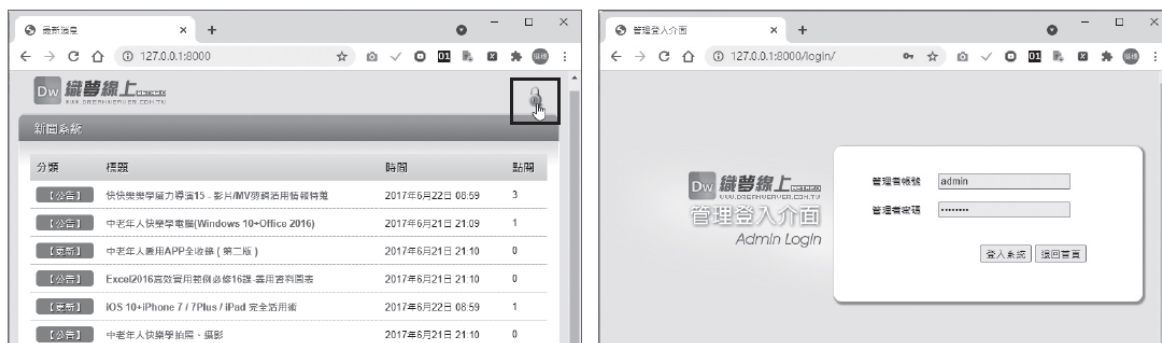
8-2 完整新聞公告系統

使用Django 內建資料庫管理功能雖然輕鬆，但每次都要開啟管理者網頁，終究不夠方便。完整新聞公告系統可自行建立資料庫管理功能，在新聞網頁中就可新增、修改及刪除新聞資料。

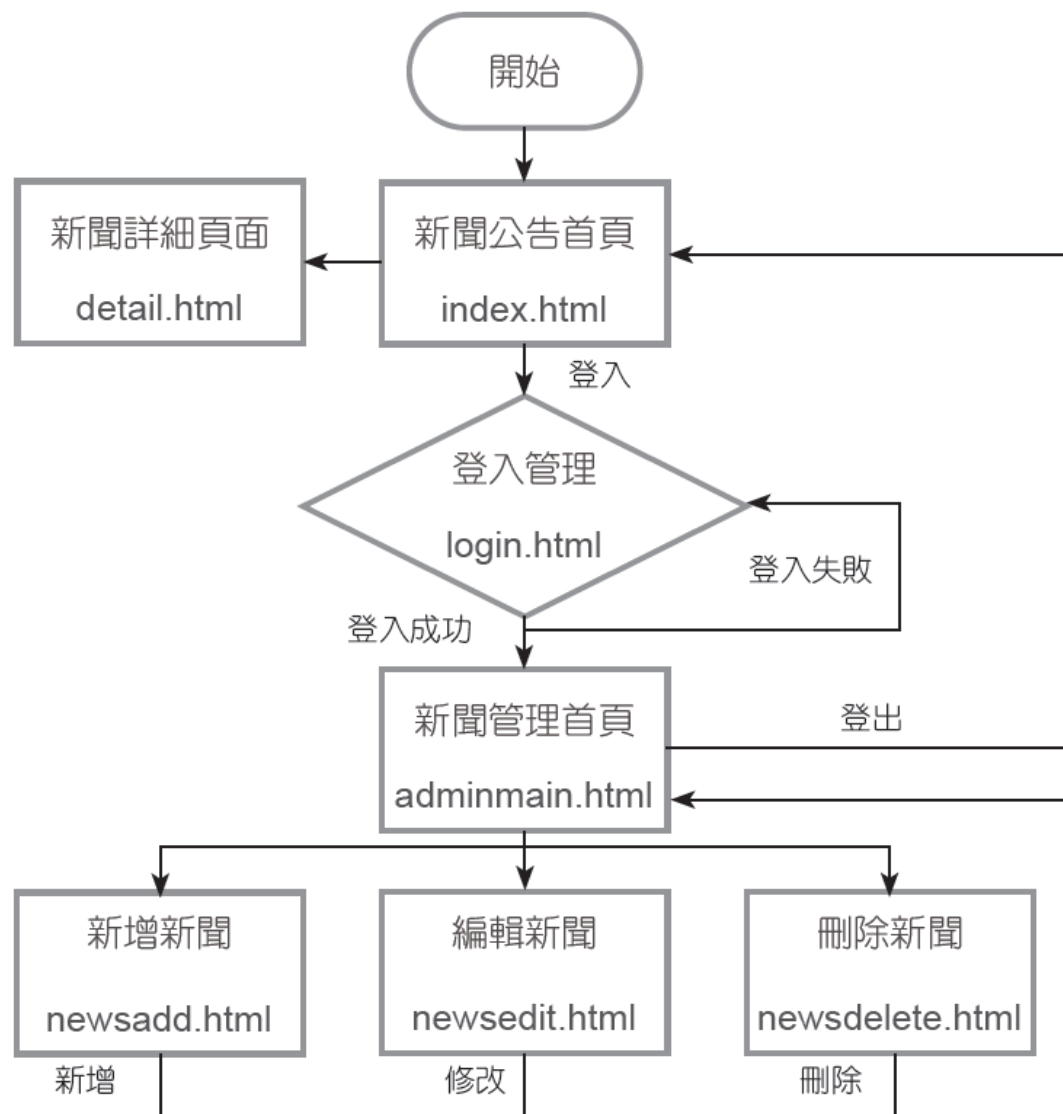
8.2.1 使用完整新聞公告系統

請將本章範例newsadm 專案複製到複製到<C:\example> 資料夾中，然後切換到newsadm 專案以「python manage.py runserver」啟動伺服器。

點選完整新聞公告系統首頁右上方  圖示即可開啟登入頁面，帳號輸入「admin」，密碼輸入「a123456!」  按 **登入系統** 鈕就可進入管理頁面。



8.2.2 完整新聞公告系統流程圖



8.2.3 Url 配置檔

newsadm\urls.py

略……

```
20 urlpatterns = [  
21     path('admin/', admin.site.urls),  
22     path('', views.index),  
23     path('index/', views.index),  
24     path('index/<str:pageindex>', views.index),  
25     path('detail/<int:detailid>', views.detail),  
26     path('login/', views.login),  
27     path('logout/', views.logout),  
28     path('adminmain/', views.adminmain),  
29     path('adminmain/<str:pageindex>', views.adminmain),  
30     path('newsadd/', views.newsadd),  
31     path('newsedit/<int:newsid>', views.newsedit),  
32     path('newsedit/<int:newsid>/<str:editttype>', views.newsedit),  
33     path('newsdelete/<int:newsid>', views.newsdelete),  
34     path('newsdelete/<int:newsid>/<str:deletetype>', views.newsdelete),  
35 ]
```

8.2.4 建立網頁基礎模版

完整新聞公告系統中<index.html> 和<detail.html> 網頁結構類似，為這2 個建立基礎模版檔<base.html>：此檔案與前一節的<base.html> 雷同，只是增加「{% blocklogin %}」以便在 <index.html> 放置登入圖示。


<adminmain.html>、<newsadd.html>、<newsedit.html> 及 <newsdelete.html> 四個檔案結構相同：標題為「最新消息 - 管理介面」，且都具有登出圖示，因此為這四個檔案建立另一個基礎模版<baseadmin.html>：




8.2.5 首頁與詳細頁面處理函式及模版

首頁處理函式(<views.py> 的index 函式)、詳細頁面處理函式(<views.py> 的detail函式) 及詳細頁面模版(<detail.html>) 皆與簡易新聞公告系統相同，只有首頁模版(<index.html>) 增加登入圖示，讓使用者可以登入管理頁面：

8.2.6 登入與登出頁面處理函式及模版

使用者在首頁點選登入管理頁面圖示  就會執行<views.py> 中的ogin 函式載入<login.html> 網頁。

使用者在管理頁面或新增、編輯、刪除新聞頁面按登出圖示  就會執行<views.py> 中的logout 函式登出管理頁面回到首頁。

8.2.7 管理頁面處理函式及模版

使用者在登入頁面通過驗證就會執行<views.py> 中的adminmain 函式載入<adminmain.html> 網頁 (管理頁面)。

adminmain 函式功能與index 函式相同，都是讀取資料及處理上一頁、下一頁功能，adminmain 函式程式碼與index 函式程式碼一樣，不再列出。


與<index.html> 相較，<adminmain.html> 在右上方多了新增新聞圖示 ，且在每則新聞右方加入編輯圖示  和刪除圖示 。




8.2.8 新增新聞頁面處理函式及模版

使用者在管理頁面點選新增新聞圖示  就會執行<views.py> 中的newsadd 函式載入<newsadd.html> 網頁。

8.2.9 編輯新聞頁面處理函式及模版

使用者在管理頁面點選新聞標題或右方編輯新聞圖示  就會執行<views.py> 中的newsedit 函式載入<newsedit.html> 網頁，讓使用者修改新聞。

8.2.10 刪除編輯新聞頁面處理函式及模版

使用者在管理頁面點選新聞右方刪除新聞圖示  就會執行<views.py> 中的newsdelete 函式載入<newsdelete.html> 網頁讓使用者刪除新聞。

<newsdelete.html> 網頁不需讓使用者輸入，只是顯示資料讓使用者確認，以免誤刪資料。



Exercise 8-0: DIV & SPAN

Div_Span.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8     <h2>DIV標記，圖片顯示分行</h2>
9     <div><img src='/static/images/DWonline.png'></div>
10    <div></div>
11
12    <h2>span標記，圖片顯示在同一行</h2>
13    <div>
14        <span></span>
15        <span></span>
16    </div>
17 </body>
18 </html>
```

Result

← → ↻ 🏠 ⓘ localhost:63342/DJ_0413/templates/

DIV標記，圖片顯示分行



span標記，圖片顯示在同一行





Exercise 8-1: 簡易新聞公告系統

urls.py

```
20 urlpatterns = [  
21     path('admin/', admin.site.urls),  
22     path('', views.index), # 不帶參數的首頁(預設第1頁)  
23     path('index/<str:pageindex>', views.index), # 帶參數的首頁  
24     path('detail/<int:detailid>', views.detail), # 詳細頁面  
25 ]
```

models.py

```
1 from django.db import models
2
3
4 class NewsUnit(models.Model): # 新聞資料表
5     catego = models.CharField(max_length=10, null=False) # 類別關聯
6     nickname = models.CharField(max_length=20, null=False) # 暱稱
7     title = models.CharField(max_length=50, null=False) # 標題
8     message = models.TextField(null=False) # 內容
9     pubtime = models.DateTimeField(auto_now=True) # 發布時間
10    enabled = models.BooleanField(default=False) # 是否顯示
11    press = models.IntegerField(default=0) # 點擊次數
12
13    def __str__(self):
14        return self.title
```

admin.py

```
1  from django.contrib import admin
2  from newsapp import models
3
4  admin.site.register(models.NewsUnit)
```

views.index (1)

```
1  from django.shortcuts import render
2  from newsapp import models
3  import math
4
5  page1 = 1  # 目前頁面
```

views.index (2)

```
8 def index(request, pageindex=None):
9     global page1 # 重複開啟本網頁時需保留 page1 的值
10    pagesize = 8 # 每頁顯示的資料筆數
11    newsall = models.NewsUnit.objects.all().order_by('-id') # 讀取新聞資料表, 依時間遞減排序
12    datasize = len(newsall) # 新聞筆數
13    totpage = math.ceil(datasize / pagesize) # 總頁數
14    if pageindex == None: # 無參數
15        page1 = 1
16        newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[:pagesize]
17    elif pageindex == '1': # 上一頁
18        start = (page1 - 2) * pagesize # 該頁第1筆資料
19        if start ≥ 0: # 有前頁資料就顯示
20            newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
21            page1 -= 1
22    elif pageindex == '2': # 下一頁
23        start = page1 * pagesize # 該頁第1筆資料
24        if start < datasize: # 有下頁資料就顯示
25            newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
26            page1 += 1
27    elif pageindex == '3': # 由詳細頁面返回首頁
28        start = (page1 - 1) * pagesize # 取得原有頁面第1筆資料
29        newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
30
31    currentpage = page1 # 將目前頁面以區域變數傳回html
32    return render(request, "index.html", locals())
33
```


views.detail (3)

```
35 def detail(request, detailid=None):
36     unit = models.NewsUnit.objects.get(id=detailid) # 根據參數取出資料
37     category = unit.catego
38     title = unit.title
39     pubtime = unit.pubtime
40     nickname = unit.nickname
41     message = unit.message
42     unit.press += 1 # 點擊數加1
43     unit.save() # 儲存資料
44
45     return render(request, "detail.html", locals())
```

views.detail (all)

```
1 from django.shortcuts import render
2 from newapp import models
3 import math
4
5 page1 = 1 # 目前頁面
6
7
8 def index(request, pageindex=None):
9     global page1 # 當使用該本網頁時保留page1 的值
10    pagesize = 8 # 每頁顯示的資料條數
11    newsall = models.NewsUnit.objects.all().order_by('-id') # 獲取新聞資料表, 按時間遞減排序
12    datasize = len(newsall) # 新聞總數
13    totpage = math.ceil(datasize / pagesize) # 總頁數
14    if pageindex == None: # 無參數
15        page1 = 1
16        newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[:pagesize]
17    elif pageindex == '1': # 上一頁
18        start = (page1 - 2) * pagesize # 該頁第1條資料
19        if start >= 0: # 有前頁資料能顯示
20            newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
21            page1 -= 1
22    elif pageindex == '2': # 下一頁
23        start = page1 * pagesize # 該頁第1條資料
24        if start < datasize: # 有下頁資料能顯示
25            newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
26            page1 += 1
27    elif pageindex == '3': # 直接跳至新聞第n頁
28        start = (page1 - 1) * pagesize # 取得第n頁第1條資料
29        newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
30
31    currentpage = page1 # 將目前頁面以區域變數存html
32    return render(request, "index.html", locals())
33
34
35 def detail(request, detailid=None):
36    unit = models.NewsUnit.objects.get(id=detailid) # 根據參數取出資料
37    category = unit.catego
38    title = unit.title
39    pubtime = unit.pubtime
40    nickname = unit.nickname
41    message = unit.message
42    unit.press += 1 # 點擊數+1
43    unit.save() # 儲存資料
44
45    return render(request, "detail.html", locals())
```

base.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset='utf-8'>
5      <title>新聞系統</title>
6      {% load static %}
7      <link href="{% static 'css/style.css' %}" rel="stylesheet" type="text/css" />
8  </head>
9  <body>
10     <div id="warp">
11         <div id="header">
12             <div class="logo">
13                 
14             </div>
15         </div>
16         <div class="headings">
17             <h2>新聞系統</h2>
18         </div>
19         {% block content %}{% endblock %}
20         <div id="footer">© Copyright 2017 eHappy Studio 新聞系統</div>
21     </div>
22 </body>
23 </html>
```


detail.html

```
1 {% extends "base.html" %}
2 {% load static %}
3 {% block content %}
4     <div class="contentbox">
5         <table width="100%">
6             <tr>
7                 <th align="left"><span class="typeblock">{{category}} </span> {{title}}</th>
8             </tr>
9             <tr>
10                <td>
11                    <span class="newsinfo">
12                         {{pubtime}}
13                         {{nickname}}
14                        <p> {{message}}</p>
15                    </span>
16                </td>
17            </tr>
18        </table>
19        <div class="topfunction">
20            <a href="/index/3/" title="返回首頁">
21                </a>
22            </div>
23    </div>
24 {% endblock %}
```

Results

| Dw 織夢線上 <small>BETA2.0</small> News Publish System | | | |
|--|----------------------|---------------------------|----|
| 分類 | 標題 | 時間 | 點閱 |
| 【test】 | test | April 12, 2022, 5:12 p.m. | 0 |
| Copyright 2022 YourSelf | | | |



Exercise 8-2: 完整新聞公告系統

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from newsadmapp import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', views.index),
23     path('index/', views.index),
24     path('index/<str:pageindex>', views.index),
25     path('detail/<int:detailid>', views.detail),
26     path('login/', views.login),
27     path('logout/', views.logout),
28     path('adminmain/', views.adminmain),
29     path('adminmain/<str:pageindex>', views.adminmain),
30     path('newsadd/', views.newsadd),
31     path('newsedit/<int:newsid>', views.newsedit),
32     path('newsedit/<int:newsid>/<str:editttype>', views.newsedit),
33     path('newsdelete/<int:newsid>', views.newsdelete),
34     path('newsdelete/<int:newsid>/<str:deletetype>', views.newsdelete),
35 ]
```


views.index (1)

```
1  import math
2  from django.contrib import auth
3  from django.contrib.auth import authenticate
4  from django.shortcuts import render, redirect
5  from newsadmapp import models
6
7  page1 = 1
```

views.index (2)

```
10 def index(request, pageindex=None): # 首頁
11     global page1
12     pagesize = 8
13     newsall = models.NewsUnit.objects.all().order_by('-id')
14     datasize = len(newsall)
15     totpage = math.ceil(datasize / pagesize)
16     if pageindex == None:
17         page1 = 1
18         newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[:pagesize]
19     elif pageindex == '1':
20         start = (page1 - 2) * pagesize
21         if start ≥ 0:
22             newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
23             page1 -= 1
24     elif pageindex == '2':
25         start = page1 * pagesize
26         if start < datasize:
27             newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
28             page1 += 1
29     elif pageindex == '3':
30         start = (page1 - 1) * pagesize
31         newsunits = models.NewsUnit.objects.filter(enabled=True).order_by('-id')[start:(start + pagesize)]
32     currentpage = page1
33     return render(request, "index.html", locals())
```

views.detail

```
36 def detail(request, detailid=None): # 詳細頁面
37     unit = models.NewsUnit.objects.get(id=detailid)
38     category = unit.catego
39     title = unit.title
40     pubtime = unit.pubtime
41     nickname = unit.nickname
42     message = unit.message
43     unit.press += 1
44     unit.save()
45     return render(request, "detail.html", locals())
```

views.login

```
48 def login(request): # 登入
49     messages = '' # 初始時清除訊息
50     if request.method == 'POST': # 如果是以POST方式才處理
51         name = request.POST['username'].strip() # 取得輸入帳號
52         password = request.POST['password'] # 取得輸入密碼
53         user1 = authenticate(username=name, password=password) # 驗證
54         if user1 is not None: # 驗證通過
55             if user1.is_active: # 帳號有效
56                 auth.login(request, user1) # 登入
57                 return redirect('/adminmain/') # 開啟管理頁面
58             else: # 帳號無效
59                 messages = '帳號尚未啟用！'
60         else: # 驗證未通過
61             messages = '登入失敗！'
62     return render(request, "login.html", locals())
```

views.logout

```
65 def logout(request): # 登出
66     auth.logout(request)
67     return redirect('/index/')

```

views.adminmain

```
70 def adminmain(request, pageindex=None): # 管理頁面
71     global page1
72     pagesize = 8
73     newsall = models.NewsUnit.objects.all().order_by('-id')
74     datasize = len(newsall)
75     totpage = math.ceil(datasize / pagesize)
76     if pageindex == None:
77         page1 = 1
78         newsunits = models.NewsUnit.objects.order_by('-id')[:pagesize]
79     elif pageindex == '1':
80         start = (page1 - 2) * pagesize
81         if start ≥ 0:
82             newsunits = models.NewsUnit.objects.order_by('-id')[start:(start + pagesize)]
83             page1 -= 1
84     elif pageindex == '2':
85         start = page1 * pagesize
86         if start < datasize:
87             newsunits = models.NewsUnit.objects.order_by('-id')[start:(start + pagesize)]
88             page1 += 1
89     elif pageindex == '3':
90         start = (page1 - 1) * pagesize
91         newsunits = models.NewsUnit.objects.order_by('-id')[start:(start + pagesize)]
92     currentpage = page1
93     return render(request, "adminmain.html", locals())
```

view.newsadd

```
96 def newsadd(request): # 新增資料
97     message = '' # 清除訊息
98     category = request.POST.get('news_type', '') # 取得輸入的類別
99     subject = request.POST.get('news_subject', '')
100     editor = request.POST.get('news_editor', '')
101     content = request.POST.get('news_content', '')
102     ok = request.POST.get('news_ok', '')
103     if subject == '' or editor == '' or content == '': # 若有欄位未填就顯示訊息
104         message = '每一個欄位都要填寫...'
105     else:
106         if ok == 'yes': # 根據ok值設定enabled欄位
107             enabled = True
108         else:
109             enabled = False
110         unit = models.NewsUnit.objects.create(catego=category, nickname=editor, title=subject, message=content,
111                                                enabled=enabled, press=0)
112         unit.save()
113         return redirect('/adminmain/')
114     return render(request, "newsadd.html", locals())
```

views.edit(1)

```
117 def newseedit(request, newsid=None, edittype=None): # 修改資料
118     unit = models.NewsUnit.objects.get(id=newsid) # 讀取指定資料
119     categories = ["公告", "更新", "活動", "其他"]
120     if edittype == None: # 進入修改頁面,顯示原有資料
121         type = unit.catego
122         subject = unit.title
123         editor = unit.nickname
124         content = unit.message
125         ok = unit.enabled
126     elif edittype == '1': # 修改完畢,存檔
127         category = request.POST.get('news_type', '')
128         subject = request.POST.get('news_subject', '')
129         editor = request.POST.get('news_editor', '')
130         content = request.POST.get('news_content', '')
131         ok = request.POST.get('news_ok', '')
132         if ok == 'yes':
133             enabled = True
134         else:
135             enabled = False
136         unit.catego = category
```


views.edit(2)

134

135

136

137

138

139

140

141

142

143



```
else:
```

```
    enabled = False
```

```
    unit.catego = category
```

```
    unit.nickname = editor
```

```
    unit.title = subject
```

```
    unit.message = content
```

```
    unit.enabled = enabled
```

```
    unit.save()
```

```
    return redirect('/adminmain/')
```

```
return render(request, "newsedit.html", locals())
```

views.edit(all)

```
117 def newseedit(request, newsid=None, edittype=None): # 修改資料
118     unit = models.NewsUnit.objects.get(id=newsid) # 讀取指定資料
119     categories = ["公告", "更新", "活動", "其他"]
120     if edittype == None: # 進入修改頁面, 顯示原有資料
121         type = unit.catego
122         subject = unit.title
123         editor = unit.nickname
124         content = unit.message
125         ok = unit.enabled
126     elif edittype == '1': # 修改完畢, 存檔
127         category = request.POST.get('news_type', '')
128         subject = request.POST.get('news_subject', '')
129         editor = request.POST.get('news_editor', '')
130         content = request.POST.get('news_content', '')
131         ok = request.POST.get('news_ok', '')
132         if ok == 'yes':
133             enabled = True
134         else:
135             enabled = False
136         unit.catego = category
137         unit.nickname = editor
138         unit.title = subject
139         unit.message = content
140         unit.enabled = enabled
141         unit.save()
142         return redirect('/adminmain/')
143     return render(request, "newseedit.html", locals())
```

views.delete

```
146 def newsdelete(request, newsid=None, deletetype=None): # 刪除資料
147     unit = models.NewsUnit.objects.get(id=newsid) # 讀取指定資料
148     if deletetype == None: # 進入刪除頁面,顯示原有資料
149         type = str(unit.catego.strip())
150         subject = unit.title
151         editor = unit.nickname
152         content = unit.message
153         date = unit.pubtime
154     elif deletetype == '1': # 按刪除鈕
155         unit.delete()
156         return redirect('/adminmain/')
157     return render(request, "newsdelete.html", locals())
```