

書本配套資源：

- <http://www.hzcourse.com/web/refbook/detail/8392/208>
- <http://www.hzcourse.com/oep/image/ueditor/jsp/upload/file/20190719/63088>, Python Flask Web 开发入门与项目实战配套资源.rar

以下為補充與修正書中未詳細解釋/錯誤的內容並解決報錯，藍色頁數表示較重要的部份。

- p. 78
將 form.py 檔案名改為 forms.py，以利其他檔案套用。
在 form.py 中，將「from flask_wtf import Form」、「class BaseLogin(Form):」中的「Form」全部改為「FlaskForm」。
- p. 85
app.py 中，新增「from forms import UploadForm」、「from werkzeug.datastructures import CombinedMultiDict」。
- p. 104
app.py 中，將「import pyMySQL」、「pyMySQL.install_as_MySQLdb()」、「conn= pyMySQL.Connect…」中的「pyMySQL」全部改為「pymysql」。
- p. 191
在 config.py 中，DB_URI 中的「MySQL+pyMySQL」全部改為小寫才能成功連接資料庫。
- p. 192
manager.py 中的 MigrateCommand 套件已不再被支援，但舊版有支援，可以安裝舊版：「pip install Flask-Migrate == 2.6.0」。
- p. 192
Error: The session is unavailable because no secret key was set. Set the secret_key on the application to something unique and secret
在 app.py 中的「app.create_app()」下新增「app.secret_key: 'super secret key'」。
- p. 193
在 models.py 中，將第 1 行「db.String(100)」中的數字設定大一點，避免 check_password function 的結果永遠是 false(因儲存在資料庫中已轉換的密碼長度與使用者輸入的密碼經轉換後的長度不同)。
- p. 194
在重新執行遷移資料庫的一系列命令前，先刪除目錄中的「Migrations」檔案，否則即使更改 Users 類中的資訊，之後執行一系列命令，資料表的結構也不會改變。
- p. 203
在 views.py 中，新增「from io import BytesIO」。
- p. 205
OSError: cannot open resource
至下方連結下載字型，解壓縮後將「chinese.stzhongs.ttf」檔案放入「…/fontTools/ttLib」中。<https://freefontsdownload.net/free-stzhongsong-font-601.htm>
在 captcha.py 中，將 function「create_validate_code」中的參數

「font_type="STZHONGS.TTF"」改為「font_type="chinese.stzhongs.ttf 的絕對路徑"」。
font_path=r'C:\Users\alenf\anaconda3\Lib\site-packages\fontTools\ttLib\chinese.stzhongs.ttf'

- p. 206
將「online=request.form.get('online')」整部份的程式碼放入「admin.views.py」檔案中，「if users:」前面。
- p. 207
「@bp.route('/test/')」以下的程式碼放入「admin.views.py」檔案中。
- p. 208
從 H-ui.admin-master 檔案夾中 index.html，放到 templates/admin 中。
- p. 209
將「import apps.cms.hooks」改為「import apps.admin.hooks」。
- p. 211
將 base.html 中的<link>標籤修改為正確的檔案路徑。
- p. 217 (p. 199)
統一將密碼長度設定為 6-20 位。
- p. 222
於 cmd 中執行「pip install -U xpinyin」，再於 views.py 中新增「from xpinyin import Pinyin」。
- p. 225
發現在 h-ui 檔案夾中找不到 public.css 和 qikoo.js 檔案，從作者提供的檔案中複製。
- [p. 225](#)
從 F12 觀察到「.mainBox h3 actionBtn」中的「padding: 0 20px;」，這就是造成 article_cat_list.html 右上角「添加分類」按鈕文字無置中的原因，在 article_cat_list.html 的第 35 行的 a 標籤中新增「style="padding-left:20px;"」，讓文字置中。
- p. 227
至網址 admin/article_cat_add/ 中填入圖 10.6 的資訊，會自動跳轉到網址 admin/article_cat_list。
- p. 233
跟 views.py 中的 def creat_cat_list() 很像，先忽略。
- p. 235
在 admin.models.py 中的 class Articles_Cat() 下新增「articles = db.relationship("Articles", lazy="dynamic")」。

並在 admin.models.py 中新增

```
article_tag = db.Table('article_tag',
    db.Column('jq_article.aid', db.Integer, db.ForeignKey('jq_article.aid'), primary_key=True),
    db.Column('jq_tag.tid', db.Integer, db.ForeignKey('jq_tag.tid'), primary_key=True))
```

在 admin.models.py 中新增

```
class Articles_Tag(db.Model):
    __tablename__ = 'jq_tag'
    tid = db.Column(db.Integer, primary_key=True, autoincrement=True) # 分类ID
    aid = db.Column(db.Integer) # 外键, 文章ID
    cat_id = db.Column(db.SmallInteger) # 属于哪个文章栏目下的tag
    tag = db.Column(db.String(40), nullable=False) # 文章标签, nullable=False 是这个
```

<https://www.maxlist.xyz/2019/11/24/flask-sqlalchemy-orm2/>

- p. 237

在 config.py 中新增「import os」。

- P. 237

在 app.admin 檔案夾中建立 ueditor.py，直接複製作者提供的程式碼 (ch10/10-4/apps/ueditor.py)。

<https://github.com/fex-team/ueditor>

<https://codeantenna.com/a/BjQijqVZpy>

<https://programmer.ink/think/simple-example-of-ueditor-for-picture-upload.html>

https://blog.csdn.net/weixin_33757911/article/details/91900863

- P. 237

在安裝的 ueditor 檔案中找不到「ueditor.all.min.js」檔案，至下載來的 H-ui 檔案夾中複製(H-ui.admin-master.zip\H-ui.admin-master\lib\ueditor\1.4.3)並貼到 static 檔案夾中。

<https://chowdera.com/2021/12/202112251138084999.html>

<https://blog.csdn.net/dxyzhbb/article/details/114290100>

- p. 238

從下載來的 H-ui 檔案夾中複製檔案 article-add.html。

- p. 238

同時將例 10-7 中的內容複製貼上到 article_list.html(p. 244)和 article_edit.html(p. 247)中。

- p. 239

在 article_add.html 中找不到「autoHeightEnabled:false」，需去作者提供的檔案中複製，如下：

```

<script type="text/plain" id="j_ueditorupload" style="height:5px;display:none;" ></script>
<script>
    //实例化编辑器
    var o_ueditorupload = UE.getEditor('j_ueditorupload',
    {
        autoHeightEnabled:false,
        'serverUrl': '/ueditor/upload/'
    });
    o_ueditorupload.ready(function ()
    {

o_ueditorupload.hide();//隐藏编辑器

//监听图片上传
o_ueditorupload.addListener('beforeInsertImage', function (t,arg)
{
    |   $("#picture").attr("value", arg[0].src);
});

```

```

/* 文件上传监听
 * 需要在ueditor.all.min.js文件中找到
 * d.execCommand("insertHtml",l)
 * 之后插入d.fireEvent('afterUpfile',b)
 */
    o_ueditorupload.addListener('afterUpfile', function (t, arg)
    {
        |   $("#picture").attr("value", arg[0].src);
    });
});

//弹出图片上传的对话框
function upImage()
{
    var myImage = o_ueditorupload.getDialog("insertimage");
    myImage.open();
}
//弹出文件上传的对话框
function upFiles()
{
    var myFiles = o_ueditorupload.getDialog("attachment");
    myFiles.open();
}
}
</script>

```

- p. 243
發現 ueditor 檔案夾中沒有 config.json，至 H-ui 檔案夾中複製(H-ui.admin-master.zip\H-ui.admin-master\lib\ueditor\1.4.3\php)並貼到 static\ueditor 中。
- p. 243
發現在瀏覽器輸入 http://127.0.0.1:5000/admin/article_add 後，頁面沒有顯示 ueditor 輸入器(如下圖)。
將 p. 237 最下面跟 ueditor 檔案相關的程式碼(第 1-2 行)，剪下貼到 article_add.html 的 head 標籤中，即可正確顯示 ueditor 輸入器。

文章標題:

簡略標題:

*分類欄目: 公司動態 ▼

關鍵字:

文章摘要: 說點什麼...至少10個字 0/200

文章作者:

文章來源:

允許評論: ☐

允許發佈: ☐

縮略圖:

文章內容:

- p. 244
在 article_list.html 新增以下程式碼。

```
<script
src="{{ url_for('static', filename='lib/datatables/1.10.15/jquery.dataTables.min
.js' ) }}"></script>
<script
src="{{ url_for('static', filename='lib/laypage/1.2/laypage.js' ) }}"></script>
```

- p. 245
至 H-ui 檔案夾中複製 article-list.html 至 templates/admin 中，並重新命名為 article_list.html。
- p. 246
將第 64 行改為「{{ row.cat_id }}」，並將第 65 行改為「{{ row.source }}」。
- p. 247
複製 p. 237 下方第 1-2 行與 ueditor 相關的檔案引入程式碼至 article_edit.html 的 head 標籤中。
- p. 247
於 views.py 新增「from sqlalchemy import func」。
- p. 247
若要成功執行 article_list 視圖，需在 views.py 中先加上 search_list(作者檔案夾中)、article_del(p. 253)與 article_all_del(p. 254)視圖，並至作者檔案夾中複製 search_list.html 至 templates/admin 中，因前述皆包含在 article_list.html 中。
- p. 251
將 views.py 中的 article_edit 視圖的第 19 行「username=' admin' 20」中的「20」刪掉。
- [p. 251](#)

werkzeug.exceptions.BadRequestKeyError: 400 Bad Request: The browser (or proxy) sent a request that this server could not understand. KeyError: 'editorValue'

表示 ueditor 編輯器沒有正確顯示，'editorValue' 是當表單提交時，編輯器內容預設的名稱。

正確引入 ueditor 編輯器相關的檔案即可解決。

- [p. 254](#)

在 article_all_del 視圖中，將第 4 行修改為

「id=list(request.values.get('aid'))」。原本的 id 類型是字串，將之改為串列，才能正確被「filter(Articles.aid.in_(id))」中的「in_」使用。

- [p. 258](#)

從作者提供的檔案中複製「admin_base.html」、

「admin_common_header.html」、「admin_common_footer.html」至 templates/admin 中。

- [p. 258](#)

將 admin_base.html 中的第 17 行修改為「{% block footer_css_js %}{% endblock %}」，才能正確運行(原本書籍打成 footr)。

- [p. 262](#)

書中沒提及 Comment，在 models.py 中新增以下程式碼，再在 cmd 執行

「python manager.py db migrate」、「python manager.py db upgrade」建立資料表。

```
#评论
class Comment(db.Model):
    __tablename__ = 'jq_comment'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    aid = db.Column(db.Integer) #外键，文章ID
    title = db.Column(db.String(255)) #评论的文章标题
    user_id = db.Column(db.Integer) #用户ID
    user_name = db.Column(db.String(200)) #用户名
    comment = db.Column(db.Text) #评论内容
    status = db.Column(db.Integer, default=0) # 当前评论状态 如果为0代表审核通过，1代表审核中，-1代表审核不通过
    parent_id = db.Column(db.Integer) #评论的父ID
    comment_ip = db.Column(db.String(255)) #评论者的IP地址
    add_time = db.Column(db.DateTime, default=datetime.now) #评论添加时间
```

- [p. 277](#)

在 admin_system_log.html 中，將第 3 行修改為「value="{{ v.id }}" name="smallBox"」，以與 p. 280 的 system_log_all_del() function 中的「input[name=smallBox]:checked」對應。

- [p. 279](#)

於 views.py 新增「from .models import Admin_log」。

- [p. 279](#)

在 views.py 中，將 admin_log_del 視圖的第 4 行，修改為「filter_by(id = id)」，才能正確運行(原先書籍是兩個等號，不符合 filter_by 參數設置)。

- p. 281

將第 16 行刪掉，刪完會自動跳轉 admin/admin_log_list，因為本來就在這個網址上操作。

- p. 284

在 Auth 類中新增以下程式碼，以符合 p. 290 資料表中呈現的欄位。

```
parent_id=db.Column(db.Integer)
status=db.Column(db.Integer)
```

- p. 285

在 Role 類中新增「description=db.Column(db.String(300))」，以與 p. 298 的 admin_add_role 視圖的第 21 行對應。

- p. 288

在 views.py 中，新增「from .models import Auth」、「from .recursion import build_auth_tree, build_auth_table」、「from .forms import Checek_Auth」。

- p. 288

至作者提供的檔案夾中，在 resursion.py 中複製 build_auth_tree, build_auth_table，在 forms.py 複製 Checek_Auth。

- p. 293

接續第 19 行，新增以下程式碼。

```
{% endblock %}

{% include "admin/admin_common_footer.html" %}

{% block footer_css_js %}

<script src="{{ url_for('static', filename='lib/My97DatePicker/4.8/WdatePicker.js') }}"></script>
<script src="{{ url_for('static', filename='lib/laypage/1.2/laypage.js') }}"></script>
```

- p. 295

執行網址/admin/admin_add_permission/，得到的頁面如下圖，發現兩個輸入框連在一起。在 admin_add_permission.html 中的 head 標籤中新增 css 「.row.cl {margin:10px 0px;）」，變成下下圖。

The screenshot shows a web form with the following elements:

- 上級分類: A dropdown menu with "無" selected.
- 權限名稱: A text input field.
- 權限URL: A text input field with "權限地址" entered.
- 是否顯示: A dropdown menu with "是" selected.
- 提交: A blue button.

- p. 296
在 admin_del_permission 視圖中，將第 5 行修改為「id=int(request.values.get('id'))」；將第 7 行修改為「…filter(Auth.id == id)…」，原先無指定 Auth 會導致在頁面刪除的 id 與實際在資料庫中被刪除的 id 不一致。
- p. 296
在 admin_permission_del function 中，將第 2 行修改為「admin_permission_del(obj, id)」。
- p. 299
從 H-ui 檔案夾中，複製「admin-role-add.html」並貼到「admin_add_role.html」。
- p. 299
在 admin_add_role.html 中，在 body 主要內容下方新增以下程式碼。

```
{% endblock %}

{% include "admin/admin_common_footer.html" %}
{% block footer_css_js %}

<script src="{{ url_for('static',filename='lib/jquery.validation/1.14.0/jquery.validate.js') }}"></script>
<script src="{{ url_for('static',filename='lib/jquery.validation/1.14.0/validate-methods.js') }}"></script>
<script src="{{ url_for('static',filename='lib/jquery.validation/1.14.0/messages_zh.js') }}"></script>
```

- p. 299
在 admin_add_role.html 中：
 1. 在「備註」前加上「*」表示必填，以跟 forms.py 中 Checek_Role 類中的 description 條件對應。
 2. 在「備註」區塊中之 input 標籤中，修改為「id="description" name="description"」，admin_add_role 視圖才能透過 request.form 獲取 description 資訊，且 ajax_post() function 才能獲取 description 資訊。
 3. 依照 p. 290 圖 12.2 中的 id，設定每一個 checkbox 的 value(原先 value 值皆為空)，ajax_post() function 才能獲取被勾選的 checkbox 的 value。
 4. 在「確定」區塊中之 button 標籤中，新增「onclick="ajax_post()"」。

5. 將 body 中的主要內容依照 p. 299 圖 2.6 的形式更改。

- p. 300

在 admin_add_role.html 中，將第 32 行與以下修改為下列程式碼，否則即使成功提交資料，也會跳出「表單驗證失敗」。

```
success:function(data){
    layer.msg('已提交成功',{icon:1, time:2000});
    let index=parent.layer.getFrameIndex(window.name);
    setTimeout("parent.layer.closeAll()", 2000);
},
error:function(data){
    layer.msg("表單驗證失敗", {icon:1, time:2000});
}
```

- p. 304

1. 至作者提供的檔案夾中找到 recursion.py 複製 creat_auth_table 類。
2. 在 views.py 中新增「from .recursion import creat_auth_table」。
3. 至作者提供的檔案夾中複製 admin_edit_role.html。

- p. 307

在 decorators.py 中，新增「from .models import Users, Role, Auth」、「from flask import request」。

- p. 308

1. 把裝飾器 admin_auth 加到各個視圖(參考 p. 290 上方圖的 url)前。
2. 在 views.py 中，新增「from .decorators import admin_auth」。

- [p. 313](#)

在 register.html 中：

1. 在第 19 行的 form 標籤中新增「method=" POST" 」，否則將被視為 GET。
2. 在第 22 行的 input 標籤中新增「name=" username" 」，以利 request 獲取輸入資料。
3. 在第 26 行的 input 標籤中新增「name=" password1" 」，同上。
4. 在第 30 行的 input 標籤中新增「name=" password2" 」，同上。
5. 在第 34 行的 input 標籤中新增「name=" email" 」，同上。

- p. 314

將 register.html 中的 36-39 行改為以下程式碼，否則 flask.flash 無法成功執行。

```

<div class="text-right">
  <span class="flash">
    {% for msg in get_flashed_messages(category_filter=['error']) %}
      {{ msg }}
    {% endfor %}

    {% for msg in get_flashed_messages(category_filter=['ok']) %}
      {{ msg }}
    {% endfor %}
  </span>
  <button class="btn btn-primary" type="submit">登錄</button>
  <button class="btn btn-danger" data-dismiss="modal">取消</button>
</div>

```

- p. 314
在 register.html 中，將第 40 行的「data-toggle」和「data-dismiss」更改為「data-bs-toggle」和「data-bs-dismiss」，否則無法聯繫到 login.html。

- p. 316
至作者提供的檔案夾中複製例 13-2 中的檔案。

- p. 317
在 front/views.py 中新增以下程式碼：

```

from flask import render_template, request, session, redirect, url_for, flash
from .models import Members
from exts import db
from .forms import RegisterForm

```

- p. 317
至作者提供的檔案夾的 front/forms.py 複製 RegisterForm 類。
- p. 321
在 views.py 中：
 1. 新增「from .forms import LoginForm」、「from datetime import timedelta」。
 2. 新增「from config import MEMBER_USER_ID」，並於 config.py 中新增「MEMBER_USER_ID= 'MEMBERREGISTER'」。
- p. 324
至作者提供的檔案夾中複製 front/recursion.py，並於 views.py 中新增「from .recursion import build_cat_tree, build_cat_table」。
- p. 326
 1. 將程式碼寫在 views.py，而非 app.py。
 2. 新增「from ..admin.models import Users, Comment」、「from datetime import datetime」。
 3. 至作者提供的檔案夾中複製 front/forms.py 中的 CommentForm 類，並在 views.py 中新增「from .forms import CommentForm」。