

# Python Object-Oriented Program with Libraries

## Unit 6: Flask Web App Development

CHAPTER 5: INTEGRATION AND DEPLOYMENT

DR. ERIC CHOU

IEEE SENIOR MEMBER



# Objectives

---

- Maintenance of a web-site:
  - Optimize the site.
  - Update, add, deletion of pages and routes.

# Optimization of the Web-site

LECTURE 1



# Step 1: Get the base.html file from the starter template package

---

1. Get the base.html and replace the original base.html with it. You may change the base.html to base\_old.html before the replacement.
2. Comment out the non-ready part.
3. Check the base\_old.html

```
<!DOCTYPE html>
<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <!-- Required meta tags -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../base_files/bootstrap.min.css" integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYXxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
  <title>Tasks</title>
</head>
<body>
  <div class="container-fluid">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <ul class="navbar-nav">
        <li class="nav-item"><a href="file:///C:/Users/ericc/Downloads/%7B%7B%20url_for(%26%39;index%26%39;)%20%7D%7D" class="nav-link">Tasks</a></li>
        <li class="nav-item"><a href="file:///C:/Users/ericc/Downloads/%7B%7B%20url_for(%26%39;add%26%39;)%20%7D%7D" class="nav-link">Add New</a></li>
      </ul>
    </nav>
    <br>
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        {% for message in messages %}
          <div class="alert alert-primary">
            {{ message }}
          </div>
        {% endfor %}
      {% endif %}
    {% endwith %}
    <br>
    {% block main %}{% endblock %}
  </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <!-- Required meta tags -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../base_files/bootstrap.min.css" integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYIXxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
  <title>Tasks</title>
</head>
<body>
  <div class="container-fluid"><!--
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <ul class="navbar-nav">
        <li class="nav-item"><a href="file:///C:/Users/ericc/Downloads/%7B%7B%20url_for(%20%7D%7D" class="nav-
link">Tasks</a></li>
        <li class="nav-item"><a href="file:///C:/Users/ericc/Downloads/%7B%7B%20url_for(%20%7D%7D" class="nav-
link">Add New</a></li>
      </ul>
    </nav>
    <br>
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        {% for message in messages %}
          <div class="alert alert-primary">
            {{ message }}
          </div>
        {% endfor %}
      {% endif %}
    {% endwith %}
    <br> -->
    {% block main %}{% endblock %}
  </div>
</body>
</html>
```

```
<h1>About this web-site</h1>
<h3>A multiple-page site</h3>
<hr>
<ul>
  <li><a href="{{ url_for('index') }}">Tasks</a></li>
  <li><a href="{{ url_for('about') }}">Add New</a></li>
</ul>
<hr>
{% block main %}
{% endblock %}
```



# Step 2: Turn on the Buttons

1. Turn on the navigation buttons
2. Install bootstrap – **pip install Flask-Bootstrap**
3. Try to see if the page works

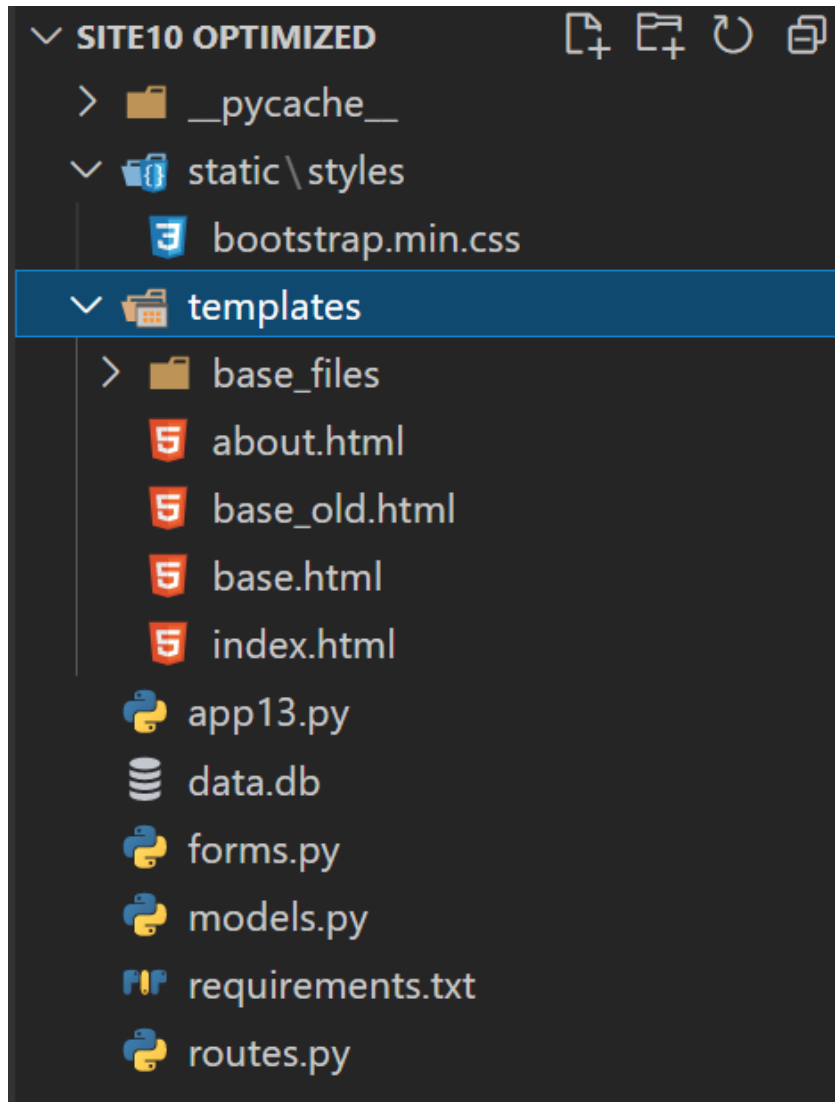
```
C:\Eric_Chou\Python Course\Python Object-Oriented Programming with Libraries\PyDev\U6 Flask\Site10 Opimized>pip install Flask-Bootstrap
Collecting Flask-Bootstrap
  Downloading https://files.pythonhosted.org/packages/88/53/958ce7c2aa26280b7fd7f3eecbf13053f1302ee2acb1db58ef32e1c23c2a/Flask-Bootstrap-3.3.7.1.tar.gz (456kB)
    100% |#####| 460kB 5.2MB/s
Requirement already satisfied: Flask>=0.8 in c:\python\python36\lib\site-packages (from Flask-Bootstrap) (1.1.2)
Collecting dominate (from Flask-Bootstrap)
  Downloading https://files.pythonhosted.org/packages/ef/a8/4354f8122c39e35516a2708746d89db5e339c867abbd8e0179bccee4b7f9/dominate-2.6.0-py2.py3-none-any.whl
Collecting visitor (from Flask-Bootstrap)
  Downloading https://files.pythonhosted.org/packages/d7/58/785fcd6de4210049da5faf62301b197f044f3835393594be368547142b0/visitor-0.1.3.tar.gz
Requirement already satisfied: Werkzeug>=0.15 in c:\python\python36\lib\site-packages (from Flask>=0.8->Flask-Bootstrap) (1.0.1)
Requirement already satisfied: click>=5.1 in c:\python\python36\lib\site-packages (from Flask>=0.8->Flask-Bootstrap) (7.1.2)
Requirement already satisfied: Jinja2>=2.10.1 in c:\python\python36\lib\site-packages (from Flask>=0.8->Flask-Bootstrap) (2.11.2)
Requirement already satisfied: itsdangerous>=0.24 in c:\python\python36\lib\site-packages (from Flask>=0.8->Flask-Bootstrap) (1.1.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\ericc\appdata\roaming\python\python36\site-packages (from Jinja2>=2.10.1->Flask>=0.8->Flask-Bootstrap) (1.0)
Installing collected packages: dominate, visitor, Flask-Bootstrap
  Running setup.py install for visitor ... done
  Running setup.py install for Flask-Bootstrap ... done
Successfully installed Flask-Bootstrap-3.3.7.1 dominate-2.6.0 visitor-0.1.3
You are using pip version 18.1, however version 21.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```



```

<!DOCTYPE html>
<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <!-- Required meta tags -->
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="../base_files/bootstrap.min.css" integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYXxHfc+NcPb1dKGj7Sk" crossorigin="anonymous">
    <title>Tasks</title>
</head>
<body>
    <div class="container-fluid">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <ul class="navbar-nav">
                <li class="nav-item"><a href="file:///C:/Users/ericc/Downloads/%7B%7B%20url_for(%20%7D%7D" class="nav-
link">Tasks</a></li>
                <li class="nav-item"><a href="file:///C:/Users/ericc/Downloads/%7B%7B%20url_for(%20%7D%7D" class="nav-
link">Add New</a></li>
            </ul>
        </nav>
        <br>
        <!--{% with messages = get_flashed_messages() %}
            {% if messages %}
                {% for message in messages %}
                    <div class="alert alert-primary">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        <br>-->
        {% block main %}{% endblock %}
    </div>
</body></html>

```



# If your external .css file is not picked up?

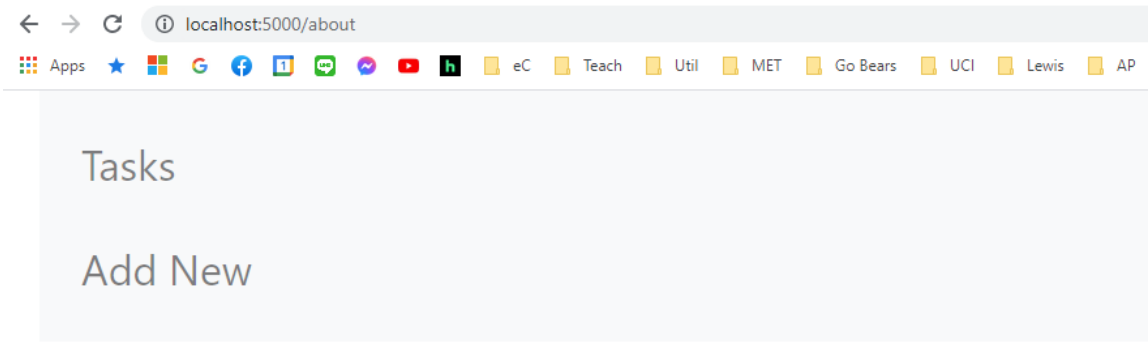
---

You need to have a 'static' folder setup (for css/js files) unless you specifically override it during Flask initialization. I am assuming you did not override it.

```

<!DOCTYPE html>
<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <!-- Required meta tags -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" type="text/css" href= "{{ url_for('static',filename='styles/bootstrap.min.css') }}">
  <title>Tasks</title>
</head>
<body>
  <div class="container-fluid">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <ul class="navbar-nav">
        <li class="nav-item"><a href="{{ url_for('index') }}" class="nav-link">Tasks</a></li>
        <li class="nav-item"><a href="{{ url_for('about') }}" class="nav-link">Add New</a></li>
      </ul>
    </nav>
    <br>
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        {% for message in messages %}
          <div class="alert alert-primary">
            {{ message }}
          </div>
        {% endfor %}
      {% endif %}
    {% endwith %}
    <br>
    {% block main %}{% endblock %}
  </div>
</body>
</html>

```



Some Content on the about Page

Title

# The Design

---

- The bootstrap.min.css takes effects after the .css file is put into the static/styles directory



## Step 3: Turn on the rest

---

1. Install Flash with `pip install Flask-Flash`
2. Modify the routes.py
3. Add flash and get\_flashed\_messages



# Downgrade Your SQLAlchemy Version

How AttributeError: can't set attribute when connecting to sqlite database with flask-sqlalchemy Error Occurs ?

Solution 2

1. Just Freeze your packages
2. `pip freeze > requirements.txt`
3. Update your SQLAlchemy in your requirements.txt
4. `SQLAlchemy<1.4.0`
5. Reinstall the packages
6. `pip install -r requirements.txt`

```
110 simpervisor==0.3
111 simplegeneric==0.8.1
112 six==1.11.0
113 SQLAlchemy==1.3.23
114 sympy==1.1.1
115 terminado==0.8.1
```



# Downgrade Your SQLAlchemy Version

```
C:\Eric_Chou\Python Course\Python Object-Oriented Programming with Libraries\PyDev\U6 Flask\Site10 Optimized>pip install SQLAlchemy==1.3.23
Collecting SQLAlchemy==1.3.23
  Downloading https://files.pythonhosted.org/packages/98/8a/d65f7914fbcbbced06303ff4f594f7cad9945af1dcfd548e1462d3814710f/SQLAlchemy-1.3.23-cp36-cp
36m-win_amd64.whl (1.2MB)
    100% |████████████████████| 1.2MB 3.1MB/s
Installing collected packages: SQLAlchemy
  Found existing installation: SQLAlchemy 1.4.19
    Uninstalling SQLAlchemy-1.4.19:
      Successfully uninstalled SQLAlchemy-1.4.19
Successfully installed SQLAlchemy-1.3.23
You are using pip version 18.1, however version 21.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

```
from flask import render_template, redirect, url_for, flash, get_flashed_messages
from app13 import app, db, Task
from datetime import datetime
import forms
```

```
@app.route('/')

```

```
@app.route('/index')
```

```
def index():
```

```
    tasks = Task.query.all()
```

```
    return render_template('index.html', tasks=tasks)
```

```
@app.route('/about', methods=['GET', 'POST'])
```

```
def about():
```

```
    form = forms.AddTaskForm()
```

```
    if form.validate_on_submit():
```

```
        t = Task(title=form.title.data, date=datetime.utcnow())
```

```
        db.session.add(t)
```

```
        db.session.commit()
```

```
        flash('Task added to the database')
```

```
        return redirect(url_for('index'))
```

```
    return render_template('about.html', form=form)
```



Tasks

Add New

Some Content on the about Page

Title

Submit

Tasks

Add New

Task added to the database

## Tasks:

- Eric Chou created on 2021-06-26
- Tommy Au created on 2021-06-26
- Nancy Brown created on 2021-06-26
- Ted Easy created on 2021-06-26
- Joyce Hong created on 2021-06-26
- Check in Hotel created on 2021-06-27
- Amber Pend created on 2021-06-27
- Serena Chen created on 2021-06-27
- Mr. X created on 2021-07-02
- Mr. Y created on 2021-07-02



Step 4: Copy the **index.html** and **add.html(about.html)** from the starter kit to current templates directory

---

- Move index.html to index\_old.html, and, then, copy index.html from starter kit to template directory
- Move about.html to about\_old.html, and, then, copy about.html from starter kit to template directory

## about\_old.html

```
<title>About</title>
{% extends "base.html" %}
{% block main %}
<p>Some Content on the about Page</p>
<form action="/about" method="post">
    {{ form.csrf_token() }}
    {{ form.title.label }}
    {{ form.title }}
    {{ form.submit }}
</form>
{% if title %}
    <h1>{{ title }}</h1>
{% endif %}
{% endblock %}
```

## about.html

```
<html>
    <head><meta http-equiv="Content-
Type" content="text/html; charset=windows-1252">
    </head>
    <body>
        {% extends "base.html" %}
        {% block main %}
            <div class="container-fluid">
                <form action="{{ url_for('about') }}" method="post">
                    {{ form.csrf_token }}
                    <div class="row">
                        <h3>Add Task</h3>
                        <div class="col">
                            {{ form.title(class="form-control") }}
                        </div>
                        <div class="col">
                            {{ form.submit(class="btn btn-primary") }}
                        </div>
                    </div>
                </form>
            </div>
        {% endblock %}
    </body>
</html>
```

Tasks Add New

Add Task

Submit

New about.html page

## index\_old.html

```
<title>Index</title>
{% extends "base.html" %}
{% block main %}
<h2>Tasks:</h2>
<ul>
    {% for task in tasks %}
    <li>{{ task.title }} created on {{ task.date }}</li>
    {% endfor %}
</ul>
{% endblock %}
```

## index.html

```
<html>
<head>
    <meta http-equiv="Content-
Type" content="text/html; charset=windows-1252">
</head>
<body>
    {% extends "base.html" %}
    {% block main %}
        {% for task in tasks %}
            <div class="card">
                <div class="card-body">
                    <h3>{{ task.title }}</h3>
                    <p>Added on {{ task.date }}</p>
                    <a href="#" class="btn btn-primary">Edit</a>
                    <a href="#" class="btn btn-danger">Delete</a>
                </div>
            </div>
        {% endfor %}
    {% endblock %}
</body>
</html>
```

Tasks Add New

## Eric Chou

Added on 2021-06-26

Edit

Delete

## Tommy Au

Added on 2021-06-26

Edit

Delete

## Nancy Brown

Added on 2021-06-26

Edit

Delete

## Ted Easy

Added on 2021-06-26

Edit

Delete

New index.html page



# Adjustment

---

- Change all about.html to add.html in Site11.
- All the occurrence of the word “about” are replaced by “add”. (Use ctrl-f to find and replace)

# Add the Edit and Delete Pages

LECTURE 2





# Step 1: Copy the **edit.html** and **delete.html** from the starter kit to current templates directory

---

Move `index.html` to `index_old.html`, and, then, copy `index.html` from starter kit to template directory

Move `about.html` to `about_old.html`, and, then, copy `about.html` from starter kit to template directory

Copy `edit.html/delete.html` to templates directory



## Step 2: Add edit page

---

- Add a `@app.route('/edit/<int:task_id>')`

```
@app.route('/edit/<int:task_id>')
def edit(task_id):
    task = Task.query.get(task_id)
    print(task)
    return redirect(url_for('index'))
```

**routes.py**



## Step 2:

---

- Update index.html (only for-loop part of the code)

```
{% for task in tasks %}
    <div class="card">
        <div class="card-body">
            <h3>{{ task.title }}</h3>
            <p>Added on {{ task.date }}</p>
            <a href="{{ url_for('edit', task_id=task.id) }}"
              class="btn btn-primary">Edit</a>
            <a href="#" class="btn btn-danger">Delete</a>
        </div>
    </div>
{% endfor %}
```

index.html

## Eric Chou

Added on 2021-06-26

Edit

Delete

## Tommy Au

Added on 2021-06-26

Edit

Delete

## Nancy Brown

Added on 2021-06-26

Edit

Delete

## Ted Easy

Added on 2021-06-26

Edit

Delete

```
* Debugger is active!
* Debugger PIN: 119-386-998
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [02/Jul/2021 00:19:03] "GET /index HTTP/1.1" 200 -
Eric Chou created on 2021-06-26
127.0.0.1 - - [02/Jul/2021 00:19:09] "GET /edit/1 HTTP/1.1" 302 -
127.0.0.1 - - [02/Jul/2021 00:19:09] "GET /index HTTP/1.1" 200 -
Nancy Brown created on 2021-06-26
127.0.0.1 - - [02/Jul/2021 00:19:15] "GET /edit/3 HTTP/1.1" 302 -
127.0.0.1 - - [02/Jul/2021 00:19:15] "GET /index HTTP/1.1" 200 -
```

Click on the buttons will cause the message to be shown



# Step 3:

---

- Create a form

```
@app.route('/edit/<int:task_id>', methods=['GET', 'POST'])
def edit(task_id):
    task = Task.query.get(task_id)
    # print(task)
    form = forms.AddTaskForm()
    if task: # if task found
        if form.validate_on_submit(): # update database on success
            task.title = form.title.data
            task.date = datetime.utcnow()
            db.session.commit()
            flash('Task has been updated.')
            return redirect(url_for('index'))
        form.title.data = task.title # keep the data base data on failure.
        return render_template('edit.html', form=form, task_id=task_id)
    return redirect(url_for('index')) # if task not found, go back to index page
```

routes.py

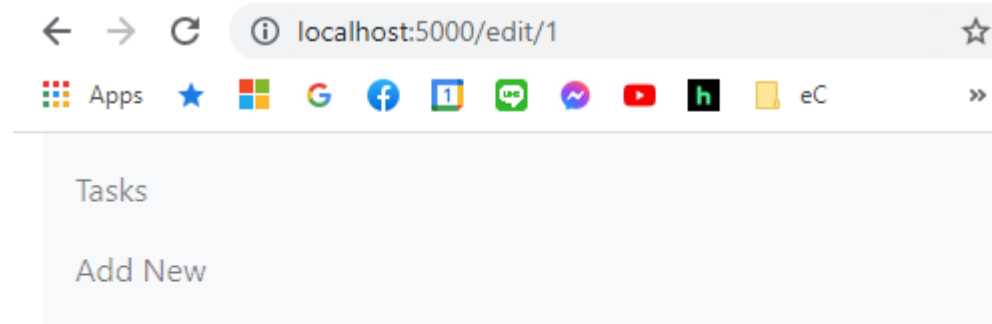
```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  </head>
  <body>
    {% extends "base.html" %}
    {% block main %}
      <div class="container-fluid">
        <form action="{{ url_for('edit', task_id=task_id) }}" method="post">
          {{ form.csrf_token }}
          <div class="row">
            <h3>Edit Task</h3>
            <div class="col">
              {{ form.title(class="form-control") }}
            </div>
            <div class="col">
              {{ form.submit(class="btn btn-primary") }}
            </div>
          </div>
        </form>
      </div>
    {% endblock %}
  </body>
</html>
```

**Eric Chou**  
Added on 2021-07-02  
[Edit](#) [Delete](#)

**Tommy Au**  
Added on 2021-06-26  
[Edit](#) [Delete](#)

**Nancy Brown**  
Added on 2021-06-26  
[Edit](#) [Delete](#)

**Ted Easy**  
Added on 2021-06-26  
[Edit](#) [Delete](#)



Edit Task

Eric Chou Jr.

Submit

Tasks Add New

Task has been updated.

**Eric Chou Jr.**  
Added on 2021-07-02  
[Edit](#) [Delete](#)

**Tommy Au**  
Added on 2021-06-26  
[Edit](#) [Delete](#)

**Nancy Brown**  
Added on 2021-06-26  
[Edit](#) [Delete](#)

**Ted Easy**  
Added on 2021-06-26  
[Edit](#) [Delete](#)



## Step 4: Add a Delete Form

forms.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import DataRequired # Guiding Rules

class AddTaskForm(FlaskForm):
    title = StringField('Title', validators=[DataRequired()])
    submit = SubmitField('Submit')

class DeleteTaskForm(FlaskForm):
    submit = SubmitField('Delete')
```





## Step 5: Add delete page

---

- Copy the `app.route()` from edit to delete

```
@app.route('/delete/<int:task_id>', methods=['GET', 'POST'])
def delete(task_id):
    task = Task.query.get(task_id)
    form = forms.DeleteTaskForm()
    if task:
        if form.validate_on_submit(): # update database on success
            db.session.delete(task)
            db.session.commit()
            flash('Task has been deleted.')
            return redirect(url_for('index'))
        return render_template('delete.html', form=form, task_id=task_id, title=task.title)
    else:
        flash('Task not found')
    return redirect(url_for('index'))
```

Tasks Add New

Eric Chou Jr.

Added on 2021-07-02

Edit

Delete

Tommy Au

Added on 2021-06-26

Edit

Delete

Nancy Brown

Added on 2021-06-26

Edit

Delete

Ted Easy

Added on 2021-06-26

Edit

Delete

Tasks

Add New

Are you sure you want to delete the following task?

Ted Easy

Delete

Cancel

Click Delete to go to  
the delete page  
For confirmation.

Ted Easy was removed

Tasks Add New

Task has been deleted.

Eric Chou Jr.

Added on 2021-07-02

Edit

Delete

Tommy Au

Added on 2021-06-26

Edit

Delete

Nancy Brown

Added on 2021-06-26

Edit

Delete

Joyce Hong

Added on 2021-06-26

Edit

Delete

# Summary

LECTURE 3



# Procedure to Learn How to Create a Flask Site

---

Step 1: Set up the flask environment and make sure the environment is valid

Use a simple program to verify it (Site 1: app.py).

Step 2: add some basic route functions (app2.py/app3.py)

Step 3: Try to render template (Site 2: app4.py/app5.py)

Use pre-defined HTML to render a HTML page. app5.py (HTML pages with parameter.

Step 4: multiple-HTML page web-site design. (Site 3: app6.py)

Step 5: add base extension for base page. (Site 4: app7.py)

Step 6: First form design (Site 5: app8.py) AddTaskForm and also separate routes.py to a separate file.



# Procedure to Learn How to Create a Flask Site

---

Step 7: Create a forms.py file and add Forms to Flask site (Site 6: app9.py)

Step 8: Handling the HTTP Post methods (Site 7: app10.py)

Step 9: connect SQLAlchemy and sqlite (Site 8: app11.py) adding database to our site.

Step 10: Update the HTML (Site 9: app12.py)

Step 11: Update the web-site presentation (Site 10: app13.py)

Step 12: Adding Edit/Delete methods to the site (Site 11: app14.py)