Cardiff's School of Computer Science & Informatics

CM1102 "Web Applications"

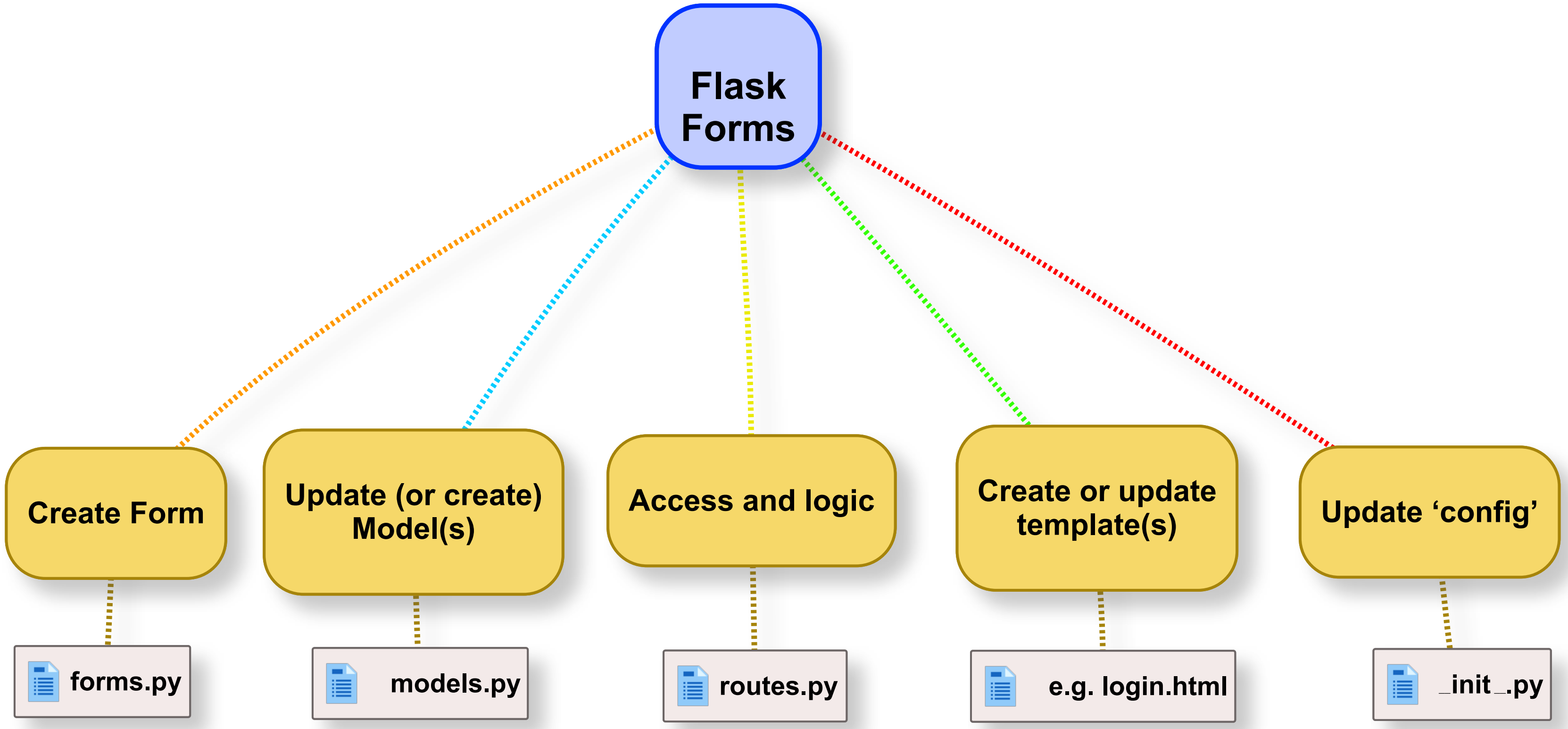
Flask 2:
Forms

Dr Natasha Edwards

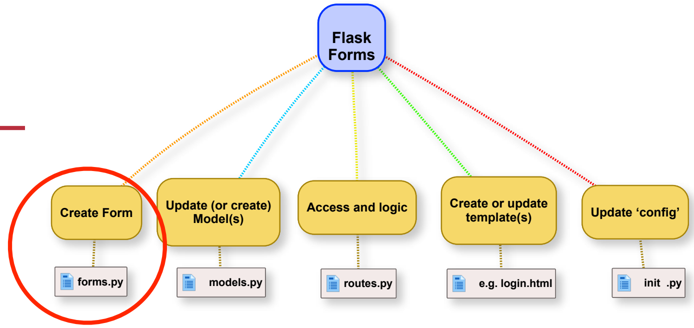
Quickstart

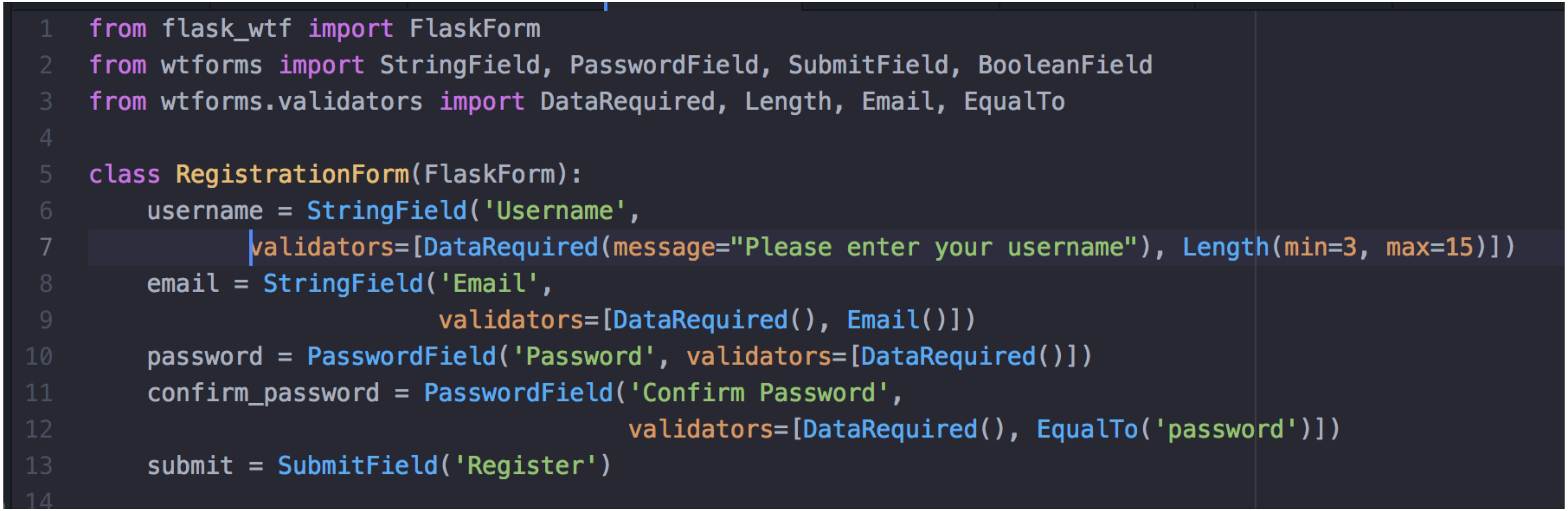
https://flask-wtf.readthedocs.io/en/stable/quickstart.html

Workflow


Flask Forms
Create Form
Update (or create) Model(s)
Access and logic
Create or update template(s)
Update ‘config’
forms.py
models.py
routes.py
e.g. login.html
init.py

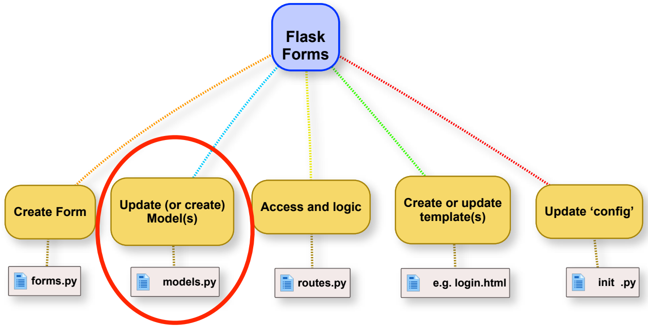
Create form

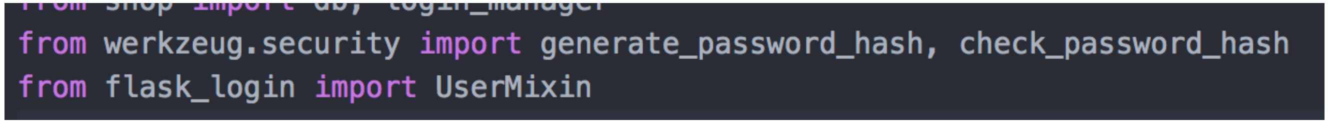




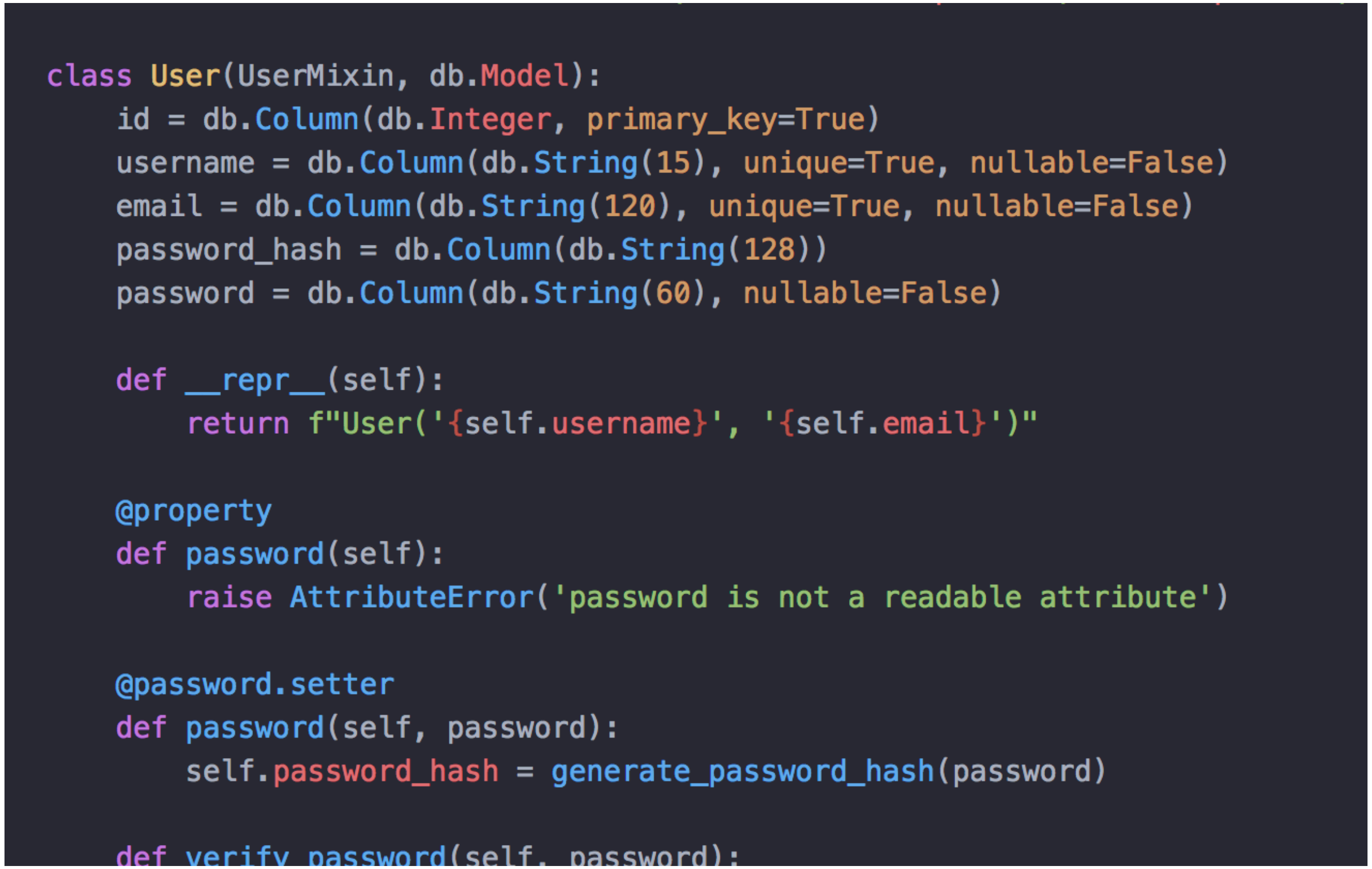
```python
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField, BooleanField
from wtforms.validators import DataRequired, Length, Email, EqualTo

class RegistrationForm(FlaskForm):
    username = StringField('Username',
           validators=[DataRequired(message="Please enter your username"), Length(min=3, max=15)])
    email = StringField('Email',
                        validators=[DataRequired(), Email()])
    password = PasswordField('Password', validators=[DataRequired()])
    confirm_password = PasswordField('Confirm Password',
                                     validators=[DataRequired(), EqualTo('password')])
    submit = SubmitField('Register')
```

Update model



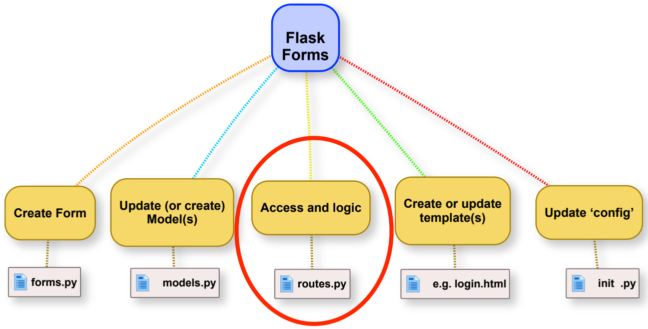


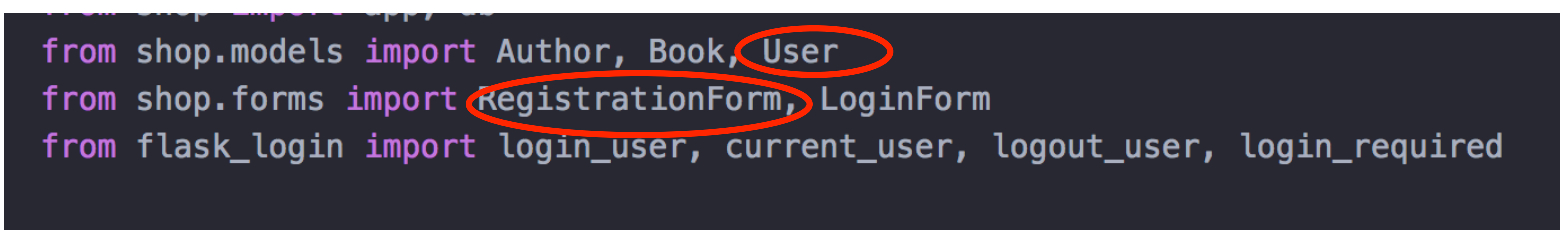
```
from shop import db, login_manager
from werkzeug.security import generate_password_hash, check_password_hash
from flask_login import UserMixin
```



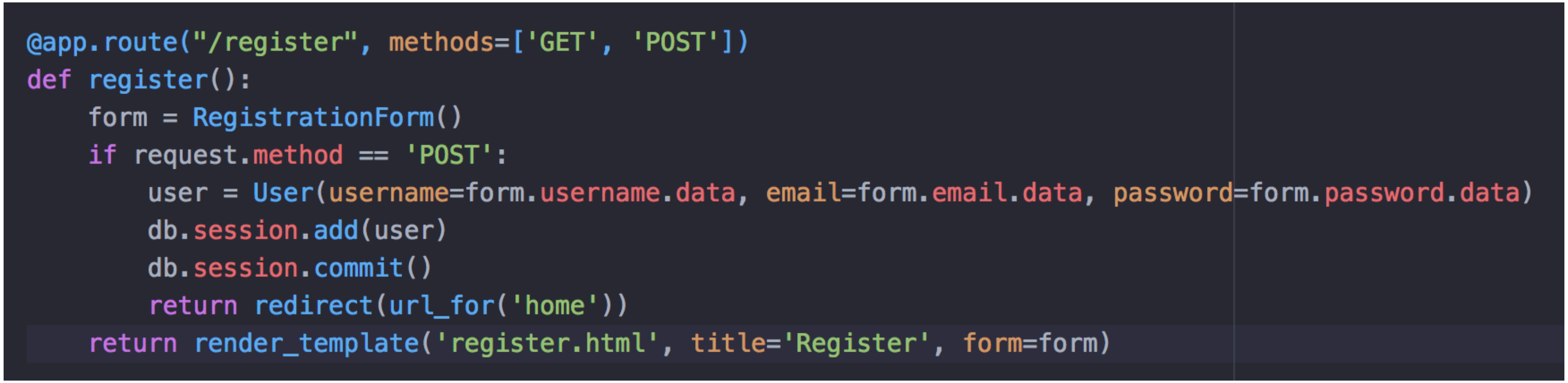
```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(15), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password_hash = db.Column(db.String(128))
    password = db.Column(db.String(60), nullable=False)

    def __repr__(self):
        return f"User('{self.username}', '{self.email}')"

    @property
    def password(self):
        raise AttributeError('password is not a readable attribute')

    @password.setter
    def password(self, password):
        self.password_hash = generate_password_hash(password)

    def verify_password(self, password):
```

Access and logic



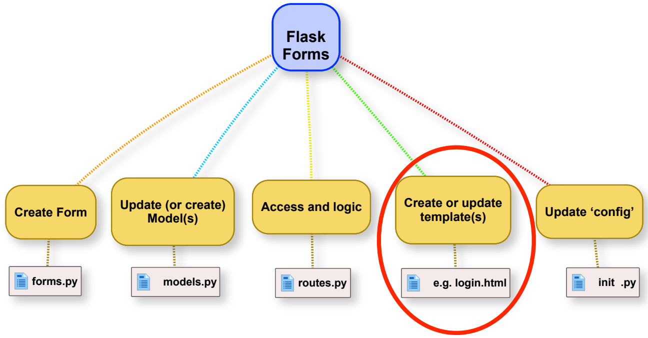


```
from shop import app, db
from shop.models import Author, Book, User
from shop.forms import RegistrationForm, LoginForm
from flask_login import login_user, current_user, logout_user, login_required
```



```
@app.route("/register", methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if request.method == 'POST':
        user = User(username=form.username.data, email=form.email.data, password=form.password.data)
        db.session.add(user)
        db.session.commit()
        return redirect(url_for('home'))
    return render_template('register.html', title='Register', form=form)
```

Create (or update) template(s)



Flask Forms

Create Form

forms.py

Update (or create) Model(s)

models.py

Access and logic

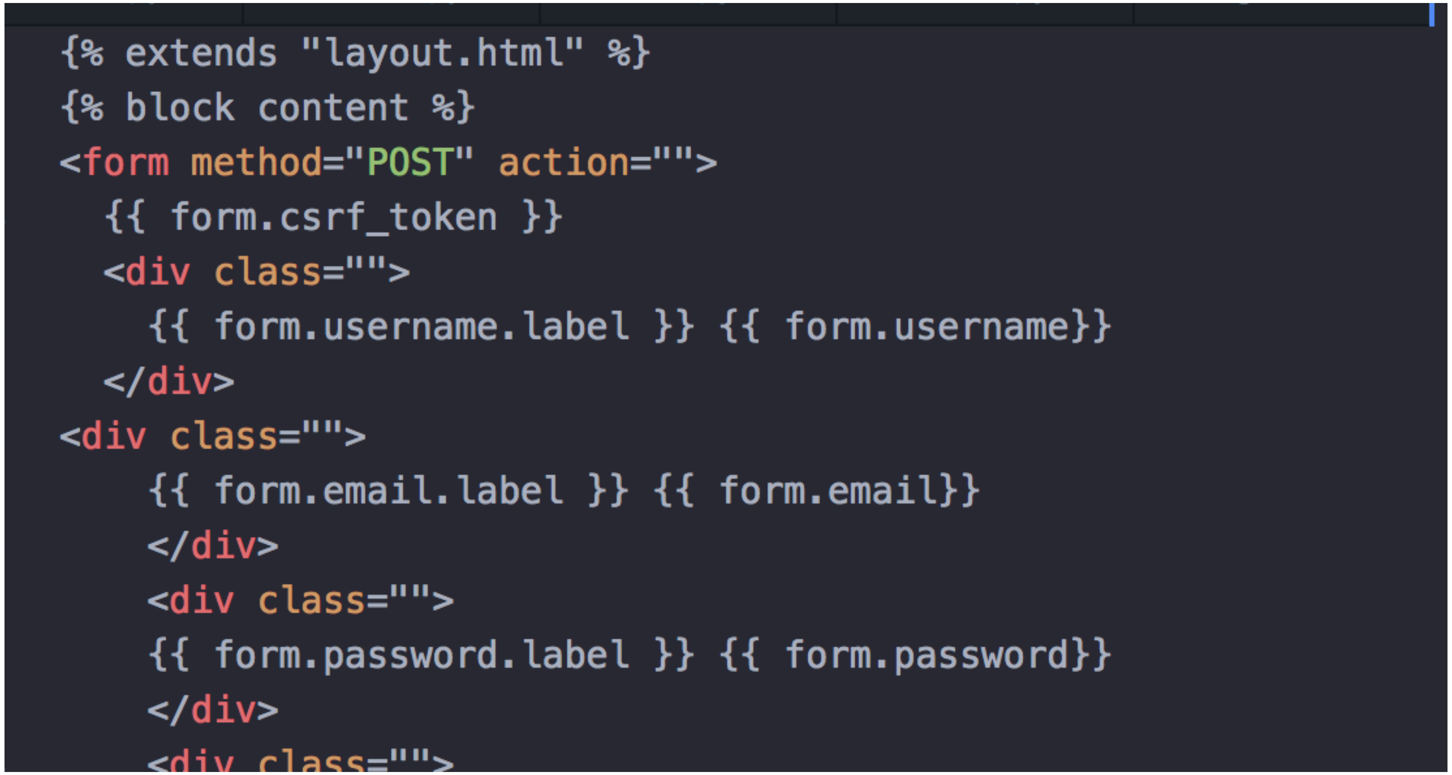
routes.py

Create or update template(s)

e.g. login.html

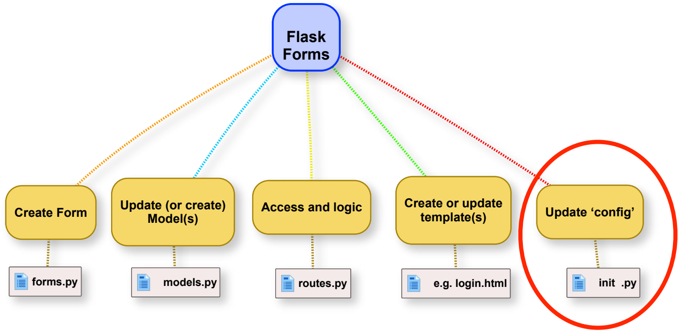
Update ‘config’

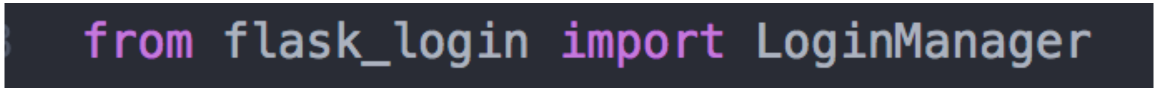
init .py



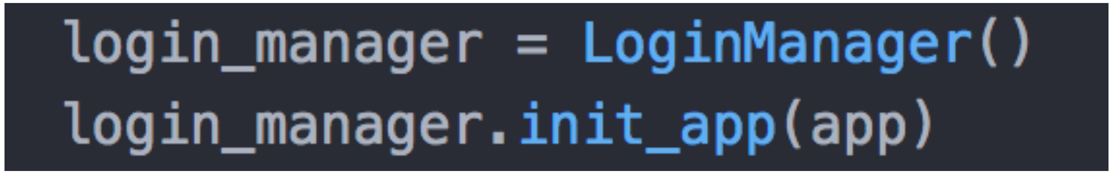
```
{% extends "layout.html" %}
{% block content %}
<form method="POST" action="">
  {{ form.csrf_token }}
  <div class="">
    {{ form.username.label }} {{ form.username}}
  </div>
<div class="">
    {{ form.email.label }} {{ form.email}}
    </div>
    <div class="">
    {{ form.password.label }} {{ form.password}}
    </div>
    <div class="">
```

Update 'config'



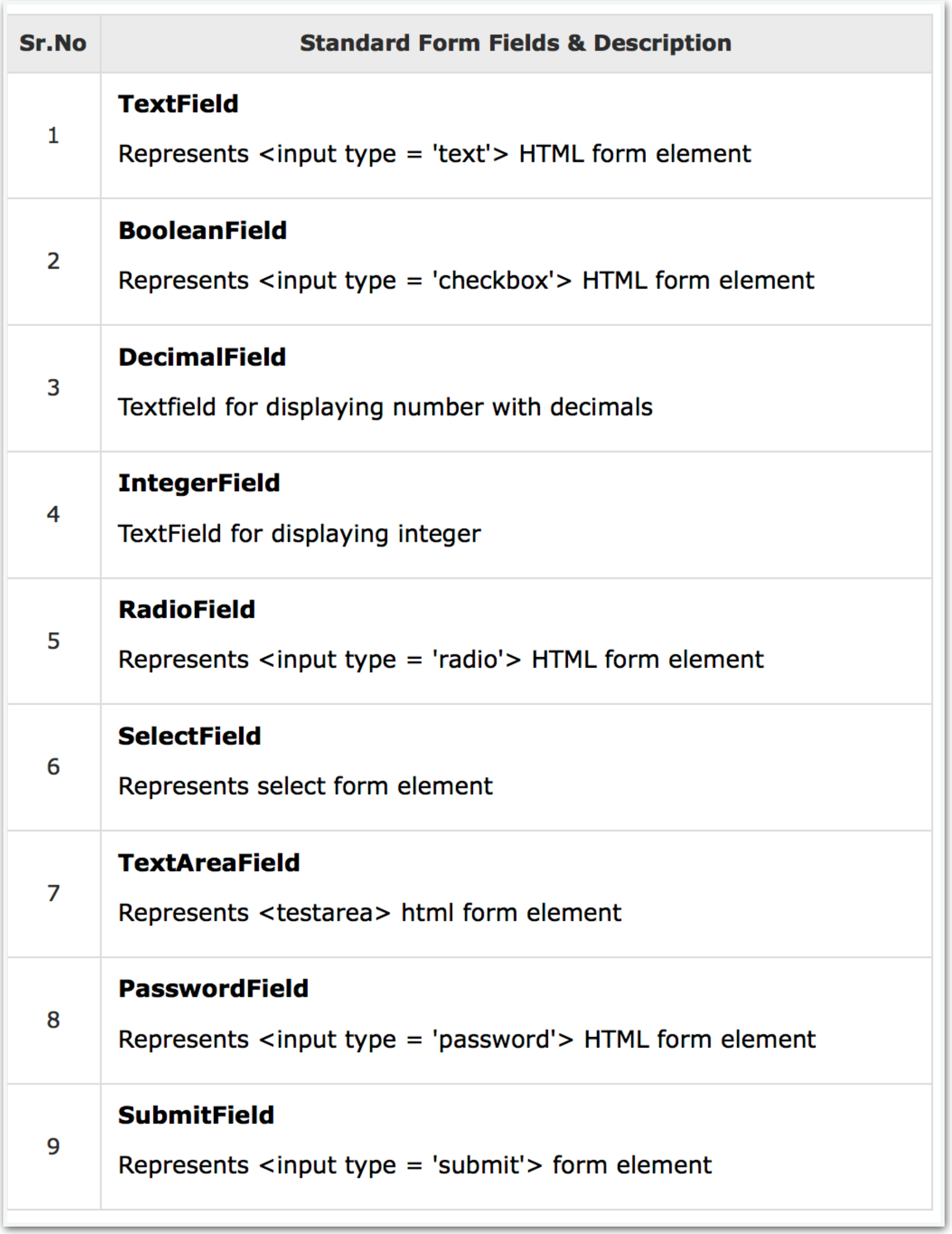


```
from flask_login import LoginManager
```



```
login_manager = LoginManager()
login_manager.init_app(app)
```

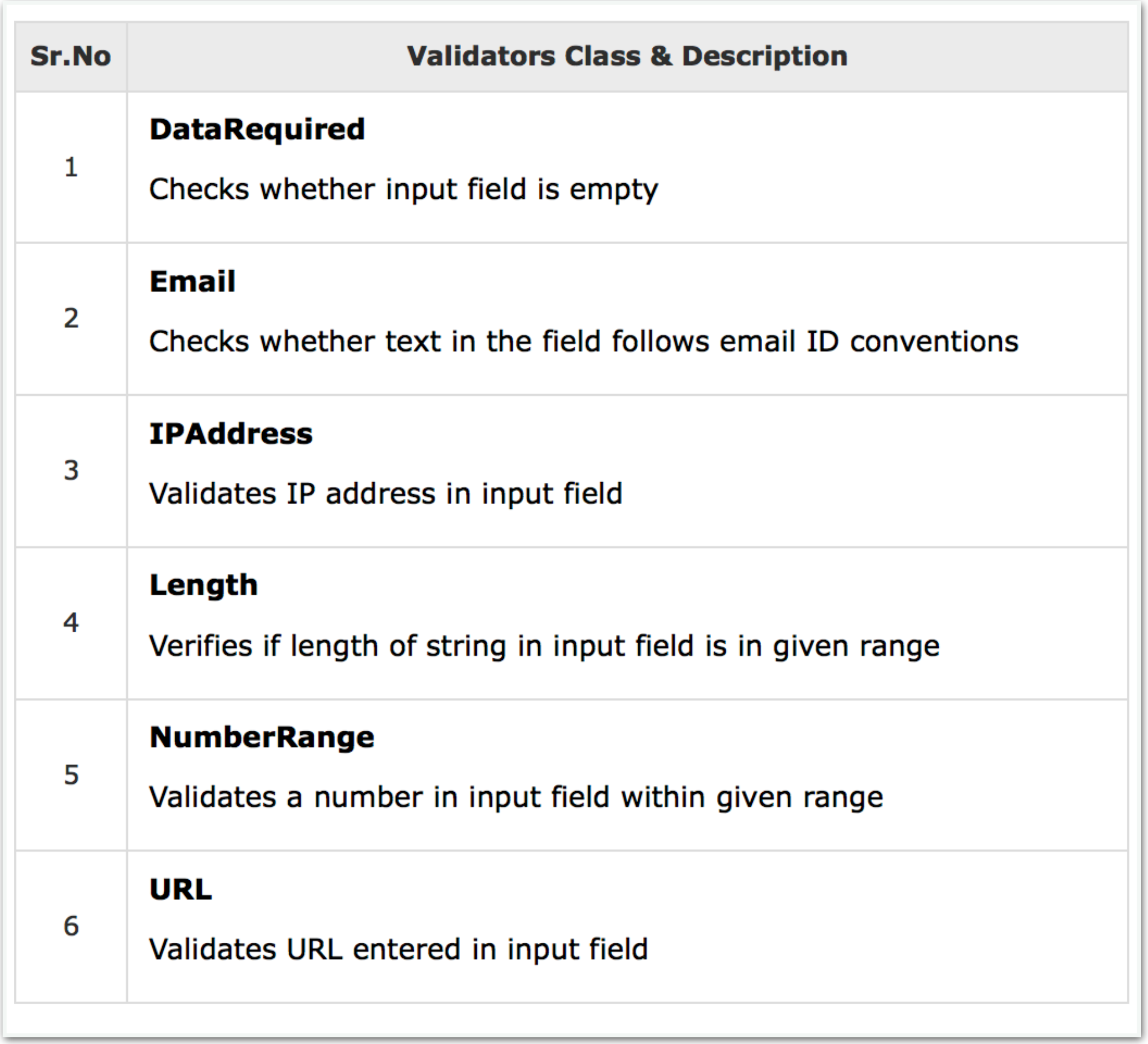
WTForms: some standard fields



Sr.No	Standard Form Fields & Description
1	**TextField** Represents <input type = 'text'> HTML form element
2	**BooleanField** Represents <input type = 'checkbox'> HTML form element
3	**DecimalField** Textfield for displaying number with decimals
4	**IntegerField** TextField for displaying integer
5	**RadioField** Represents <input type = 'radio'> HTML form element
6	**SelectField** Represents select form element
7	**TextAreaField** Represents <testarea> html form element
8	**PasswordField** Represents <input type = 'password'> HTML form element
9	**SubmitField** Represents <input type = 'submit'> form element

Source: https://www.tutorialspoint.com/flask/flask_wtf.htm

WTForms: validators



Sr.No	Validators Class & Description
1	**DataRequired** Checks whether input field is empty
2	**Email** Checks whether text in the field follows email ID conventions
3	**IPAddress** Validates IP address in input field
4	**Length** Verifies if length of string in input field is in given range
5	**NumberRange** Validates a number in input field within given range
6	**URL** Validates URL entered in input field

Source: https://www.tutorialspoint.com/flask/flask_wtf.htm