

#Barebones App

```
# importing Flask
from flask import Flask, render_template
# importing the flask_wtf library
from flask_wtf import FlaskForm
# importing the Fields
from wtforms import StringField, PasswordField,
SubmitField
# importing the validations
from wtforms.validators import DataRequired
```

Getting Started

```
app = Flask(__name__)
app.config["SECRET_KEY"] = "This is a secret key"
class LoginForm(FlaskForm):
    email = StringField(label="Email",
                        validators=[DataRequired(),
                                   validate_email()])
    submit = SubmitField(label="login")
@app.route("/")
def login():
    login_form = LoginForm()
    login_form.validate_on_submit()
    return render_template("login.html",
                           form=login_form)
```

label

```
email = StringField(label="Email")
# the label property is used in login.html when the
form object is passed over.
<form method="POST" action="{{ url_for('login') }}">
  {{ form.csrf_token }}
  {{ form.email.label }} {{ form.email(size=30) }}
  {{ form.submit }}
</form>
```

Validation

Error handling

```
# When a form is submitted, there may be a number
of errors, so a List of errors can be generated
and passed over to our form HTML as a property on
the field which generated the error, e.g.
form.<field>.errors
{{ form.email.label }} <br> {{ form.email
(size=30) }}
{% for err in form.email.errors %}
<span style="color:red">{{ err }}</span>
{% endfor %}
# Form validation
# to validate the user's entry when they hit
submit. so we have to edit our route and make sure
it is able to respond to POST requests and then to
validate_on_submit() e.g.
login_form.validate_on_submit()
```

#Template Inheritance Using Jinja2

#1. This line of code tells the templating engine (Jinja) to use "base.html" as the template for this page

```
{% extends "base.html" %}
```

#2. This block inserts a custom title into the header of the template.

```
{% block title %}Success{% endblock %}
```

#3. This block provides the content of the website. The part that is going to vary between webpages.

```
{% block content %}
  <div class="container">
    <h1>Top Secret </h1>
    <iframe src="https://giphy.com/embed/Ju7l5y9osyymQ" width="480" height="360" frameborder="0" class="giphy-embed" allowfullscreen></iframe>
    <p><a href="https://giphy.com/gifs/riick-astley-Ju7l5y9osyymQ">via GIPHY</a></p>
  </div>
{% endblock %}
```

label

```
email = StringField(label="Email")
# the label property is used in login.html when the
form object is passed over.
<form method="POST" action="{{ url_for('login') }}">
  {{ form.csrf_token }}
  {{ form.email.label }} {{ form.email(size=30) }}
  {{ form.submit }}
</form>
```

```
email = StringField(Label='Email', validators=
[DataRequired()])
# The validators parameter accepts a List of
validator Objects. DataRequired makes the field
required fields, so the user must type something,
otherwise an error will be generated.
# In order to make sure that we are giving all
users field validation, we have to switch off the
browser validation, and we do that with an
attribute on the form element called novalidate.
<form method= " POST" action="{{ url_for(' -
login') }}" novalidate>
```



By **khalidhariry**

cheatography.com/khalidhariry/

Not published yet.

Last updated 21st November, 2022.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>