LECTURE 21: USER LOGINS

Code based on:
https://github.com/miguelgrinberg/microblog/tree/v0.5

OUTLINE

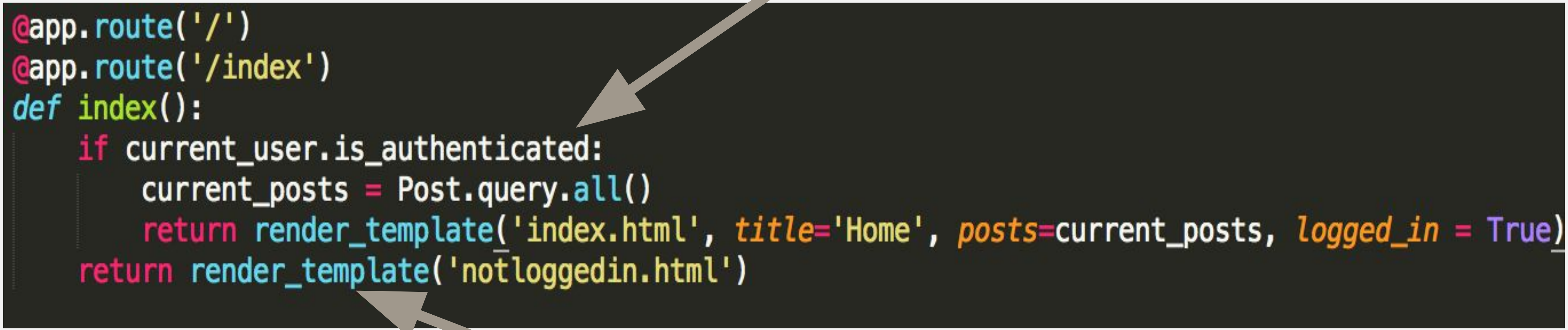
- What is a Wiki?
- Implementation
  - Displaying wiki pages
  - Submit pages
  - Index page listing wiki pages
- How about user authentication?

THIS LECTURE

- How do we implement user authentication?
- Cookies?
- Flask provides an abstraction (flask-login)
- pip3 install flask-login

HOW DO YOU CHECK FOR LOGINS?

Is user logged-in?



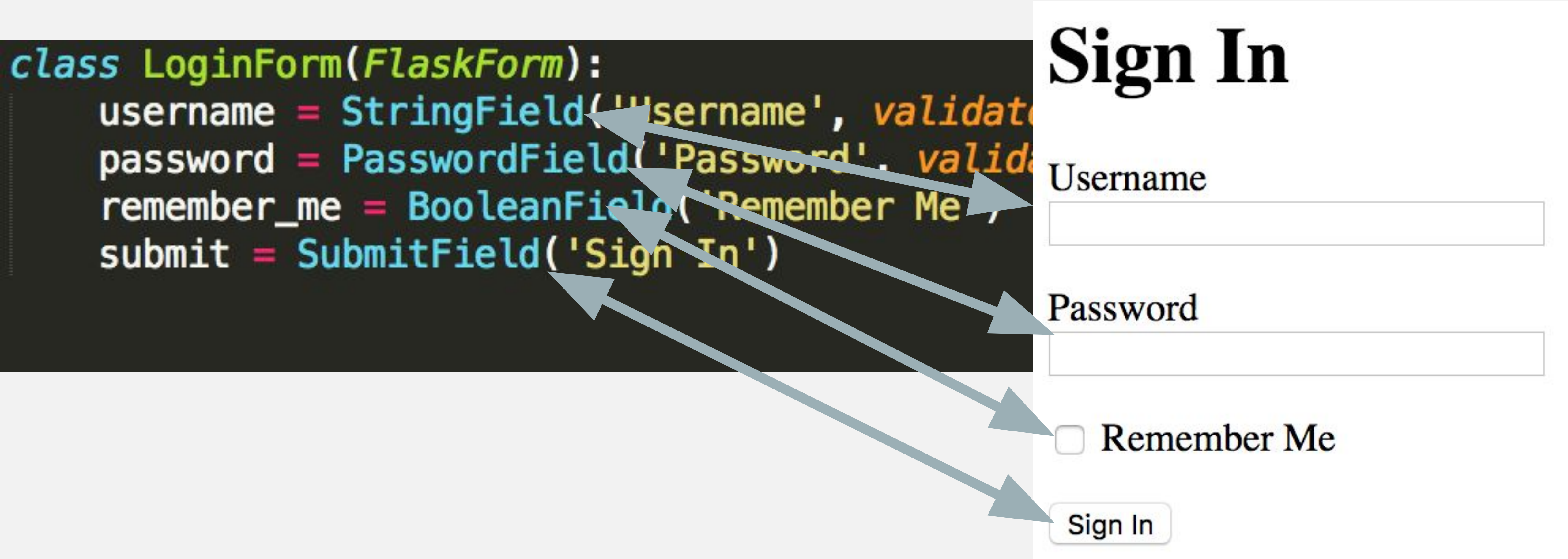
```python
@app.route('/')
@app.route('/index')
def index():
    if current_user.is_authenticated:
        current_posts = Post.query.all()
        return render_template('index.html', title='Home', posts=current_posts, logged_in = True)
    return render_template('notloggedin.html')
```

Show this if not logged-in

EXTRA FORMS?

- In addition to SubmissionForm, we have LoginForm and RegistrationForm
- LoginForm: Form users use when logging in
- RegistrationForm: Form visitors use to register

LOGIN FORM



```
class LoginForm(FlaskForm):
    username = StringField('Username', validat
    password = PasswordField('Password', valida
    remember_me = BooleanField('Remember Me')
    submit = SubmitField('Sign In')
```

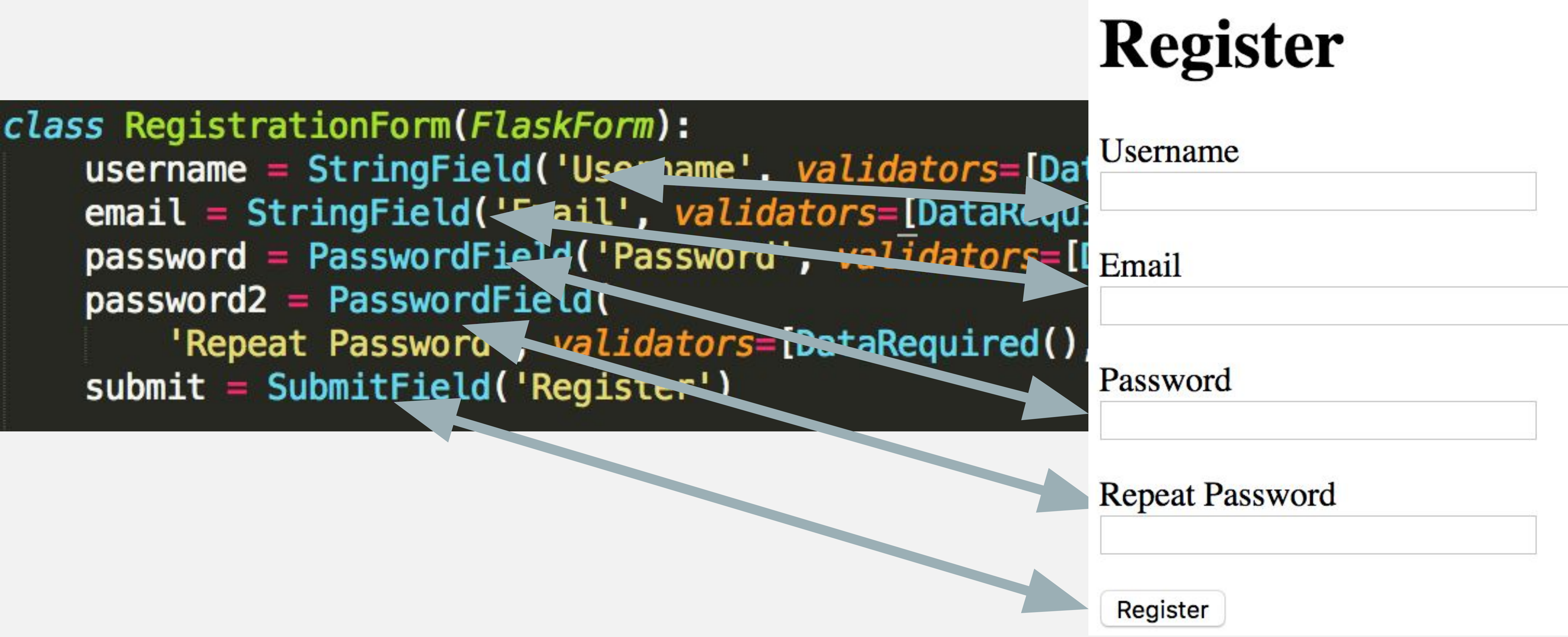
Sign In

Username

Password

☐ Remember Me

REGISTRATION FORM



```
class RegistrationForm(FlaskForm):
    username = StringField('Username', validators=[Dat
    email = StringField('Email', validators=[DataRequi
    password = PasswordField('Password', validators=[D
    password2 = PasswordField(
        'Repeat Password', validators=[DataRequired(),
    submit = SubmitField('Register')
```

Register

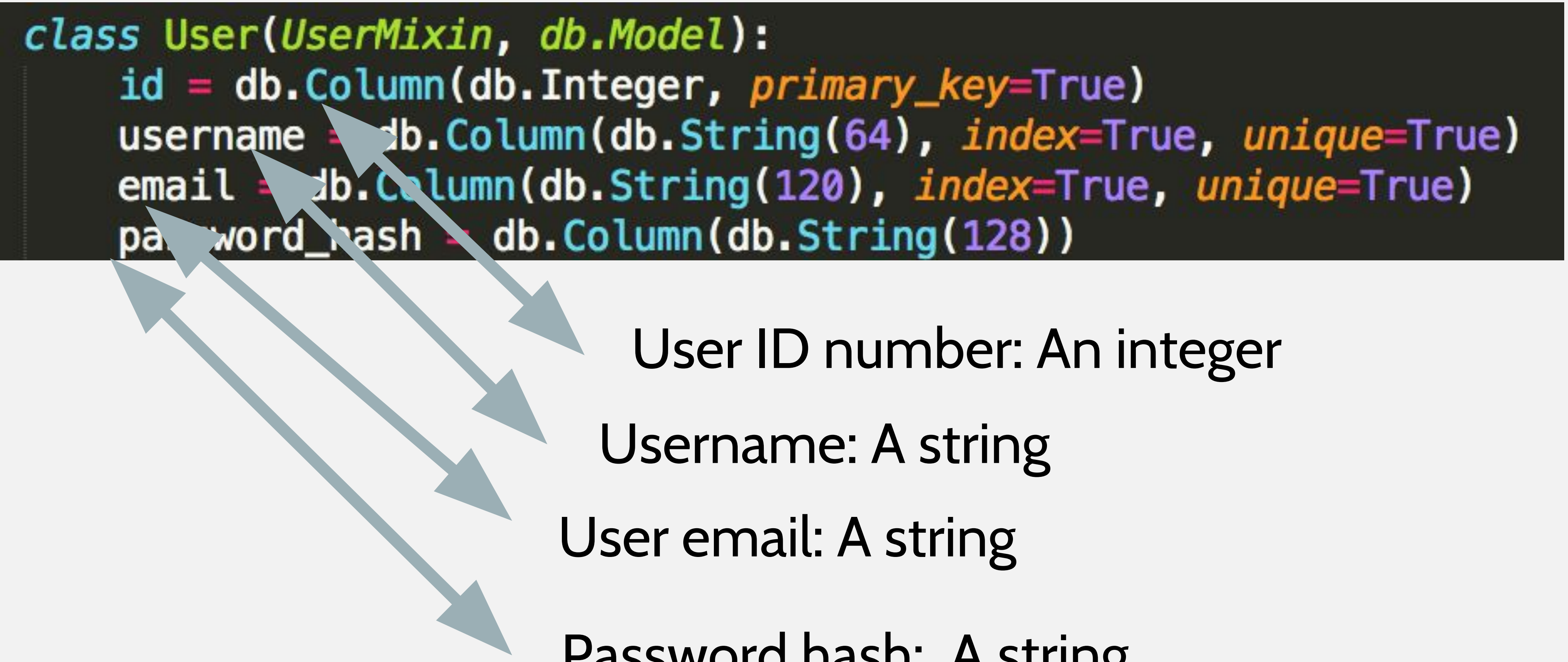
Username

Email

Password

Repeat Password

EXTRA MODEL: USER



```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(64), index=True, unique=True)
    email = db.Column(db.String(120), index=True, unique=True)
    password_hash = db.Column(db.String(128))
```

User ID number: An integer

Username: A string

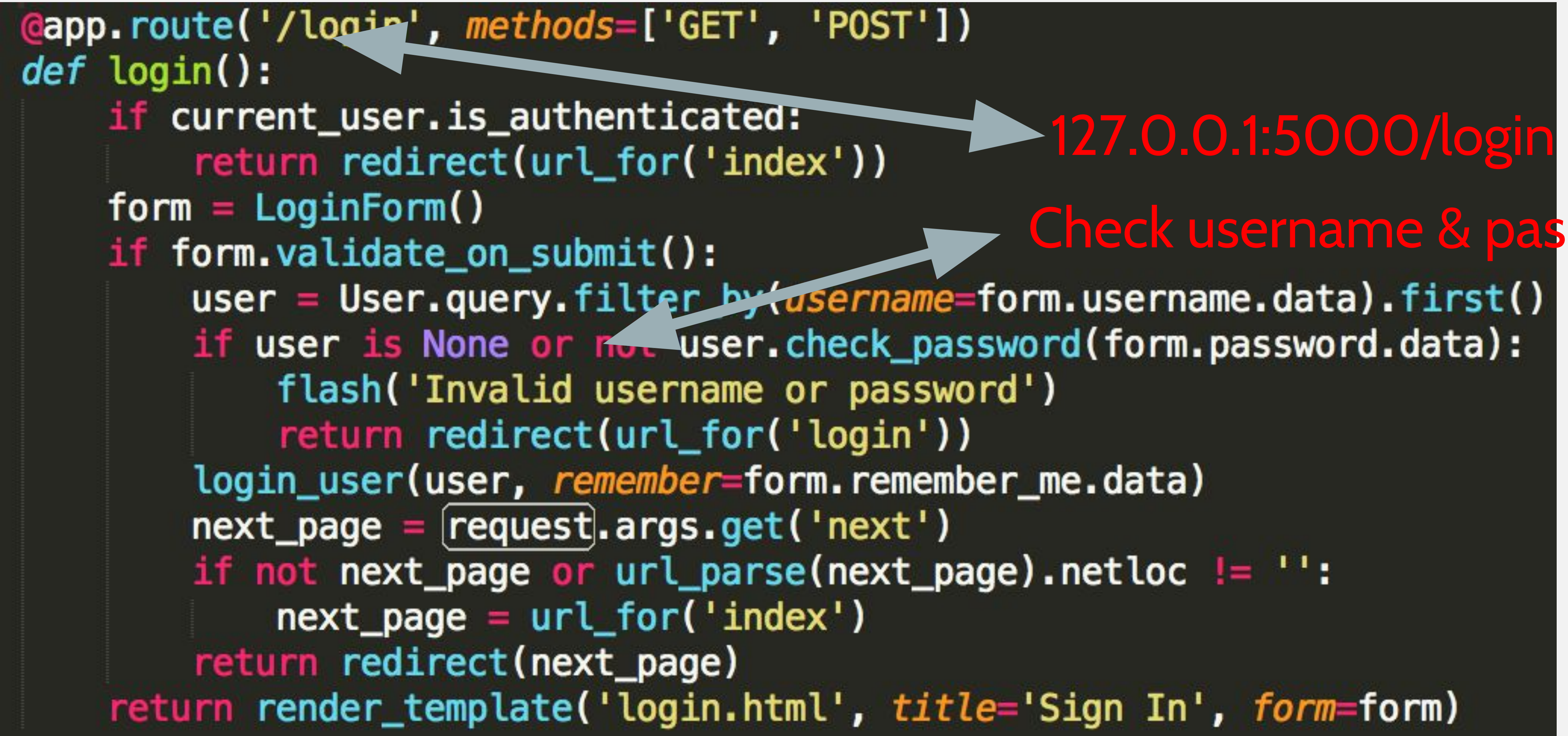
User email: A string

Password hash: A string

EXTRA PAGES?

- Login page: The login form
- Logout page: The logout operations
- Register page: The registration form

LOGIN PAGE

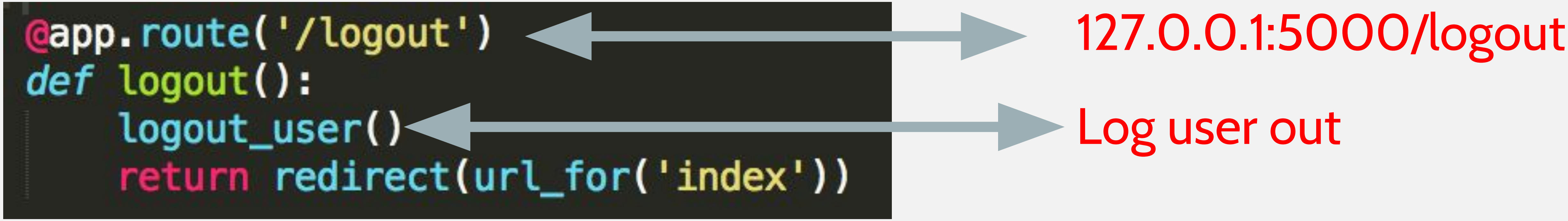


```python
@app.route('/login', methods=['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user is None or not user.check_password(form.password.data):
            flash('Invalid username or password')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
        next_page = request.args.get('next')
        if not next_page or url_parse(next_page).netloc != '':
            next_page = url_for('index')
        return redirect(next_page)
    return render_template('login.html', title='Sign In', form=form)
```

127.0.0.1:5000/login

Check username & password

LOGOUT PAGE

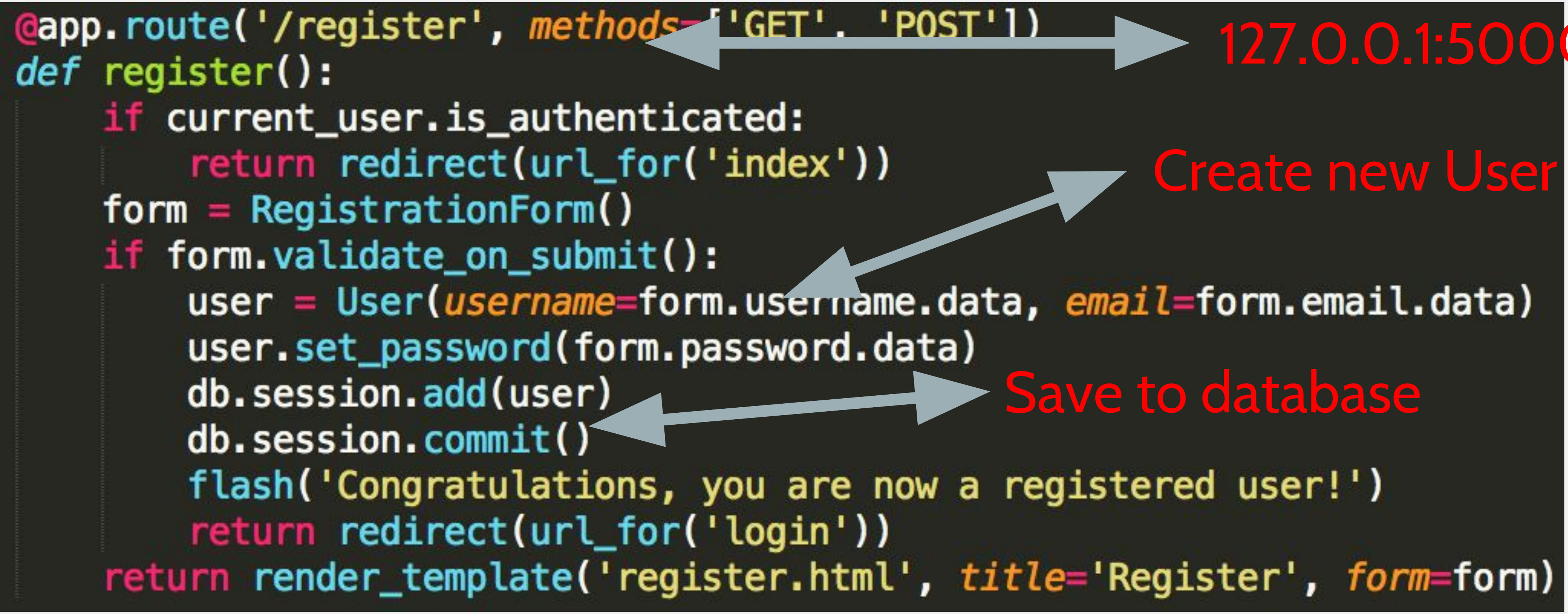


```
@app.route('/logout')
def logout():
    logout_user()
    return redirect(url_for('index'))
```

127.0.0.1:5000/logout

Log user out

REGISTER PAGE



```python
@app.route('/register', methods=['GET', 'POST'])
def register():
    if current_user.is_authenticated:
        return redirect(url_for('index'))
    form = RegistrationForm()
    if form.validate_on_submit():
        user = User(username=form.username.data, email=form.email.data)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        flash('Congratulations, you are now a registered user!')
        return redirect(url_for('login'))
    return render_template('register.html', title='Register', form=form)
```

127.0.0.1:5000/register

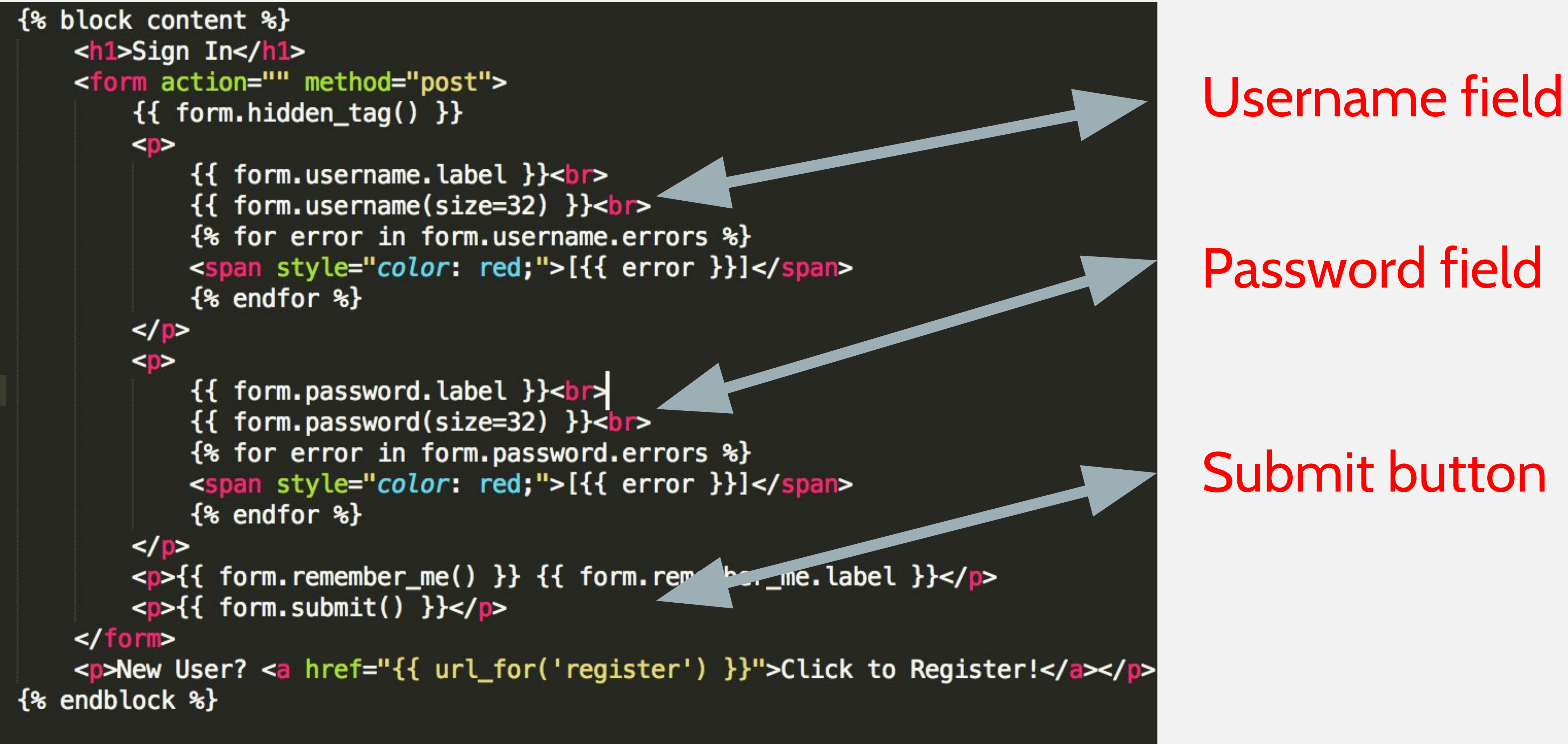
Create new User object

Save to database

EXTRA TEMPLATES

- login.html: The login page
- register.html: The registration page
- notloggedin.html: Warning for users not logged-in

LOGIN.HTML



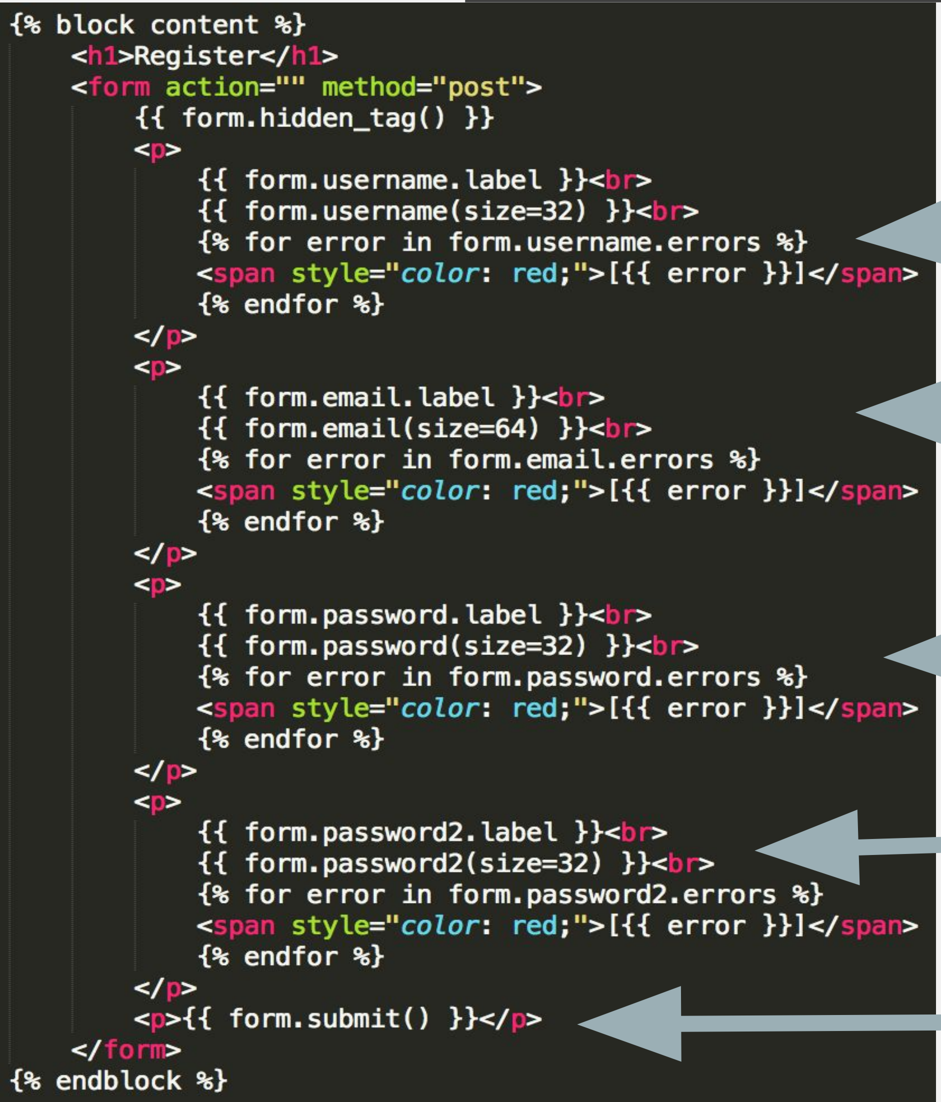
```
{% block content %}
    <h1>Sign In</h1>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        <p>
            {{ form.username.label }}<br>
            {{ form.username(size=32) }}<br>
            {% for error in form.username.errors %}
            <span style="color: red;">[{{ error }}]</span>
            {% endfor %}
        </p>
        <p>
            {{ form.password.label }}<br>
            {{ form.password(size=32) }}<br>
            {% for error in form.password.errors %}
            <span style="color: red;">[{{ error }}]</span>
            {% endfor %}
        </p>
        <p>{{ form.remember_me() }} {{ form.remember_me.label }}</p>
        <p>{{ form.submit() }}</p>
    </form>
    <p>New User? <a href="{{ url_for('register') }}">Click to Register!</a></p>
{% endblock %}
```

Username field

Password field

Submit button

REGISTER.HTML



```
{% block content %}
    <h1>Register</h1>
    <form action="" method="post">
        {{ form.hidden_tag() }}
        <p>
            {{ form.username.label }}<br>
            {{ form.username(size=32) }}<br>
            {% for error in form.username.errors %}
            <span style="color: red;">[{{ error }}]</span>
            {% endfor %}
        </p>
        <p>
            {{ form.email.label }}<br>
            {{ form.email(size=64) }}<br>
            {% for error in form.email.errors %}
            <span style="color: red;">[{{ error }}]</span>
            {% endfor %}
        </p>
        <p>
            {{ form.password.label }}<br>
            {{ form.password(size=32) }}<br>
            {% for error in form.password.errors %}
            <span style="color: red;">[{{ error }}]</span>
            {% endfor %}
        </p>
        <p>
            {{ form.password2.label }}<br>
            {{ form.password2(size=32) }}<br>
            {% for error in form.password2.errors %}
            <span style="color: red;">[{{ error }}]</span>
            {% endfor %}
        </p>
        <p>{{ form.submit() }}</p>
    </form>
{% endblock %}
```

Username field

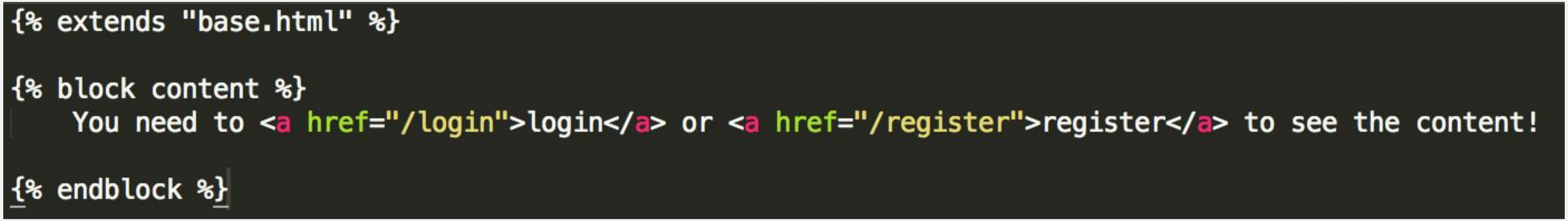
Email field

Password field

Repeat password

Submit button

NOTLOGGEDIN.HTML



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

{% block content %}
    You need to <a href="/login">login</a> or <a href="/register">register</a> to see the content!

{% endblock %}
```

FIN!