**Информационные системы управления корпоративным контентом**

**Практическая работа №14**

**ФИО:** Гусев Тимофей Андреевич

**Группа:** ИКБО-08-21

**Дата предоставления в СДО:**

**Задание 1. Дополнить веб-приложение с использованием фреймворка flask и БД sqlite - упрощенная workflow-система:**

**- при создании шаблона workflow: - загрузка docx-шаблона документа;**

**- при создании экземпляра workflow:**

**- заполненные поля могут использоваться для заполнения docx-шаблона;**

**- отправка письма адресату с вложением - заполненный docx-шаблон.**

На листинге 1 измененный представлен код функции создания шаблона.

Листинг 1 – Создание шаблона

@app.route('/new', methods=["POST", "GET"])  
def new\_item():  
 error = None  
 con = get\_db\_connection()  
 if request.method == 'POST':  
 skills = request.form.getlist('skill[]')  
 res = ""  
 title = ""  
 if skills[0] == "":  
 error = 'Задайте название'

else:  
 i = 0  
 print(skills)  
 for value in skills:  
 if i == 0:  
 title = value  
 else:  
 res += value + ","  
 i += 1  
 res = res[:-1]  
 number = get\_number(con) + 1  
 if 'file' not in request.files:  
 flash('Не могу прочитать файл')  
 return redirect(request.url)  
 file = request.files['file']  
 if file.filename == '':  
 flash('Нет выбранного файла')  
 return redirect(request.url)  
 if file and allowed\_file(file.filename):  
 filename = secure\_filename(file.filename)  
 file.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename))  
 insert\_to\_db(con, (number, title, res, filename))  
 return redirect(url\_for('index'))  
 return render\_template('new\_item.html', error=error)

На листинге 2 представлен код страницы создания шаблона.

Листинг 2 – Разметка страницы создания шаблона

{% extends 'base.html' %}  
  
{% block content %}  
 <!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />  
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  
</head>  
<body>

<script>  
$(document).ready(function() {  
  
var MaxInputs = 8; //maximum input boxes allowed  
var InputsWrapper = $("#InputsWrapper"); //Input boxes wrapper ID  
var AddButton = $("#AddMoreFileBox"); //Add button ID  
  
var x = InputsWrapper.length; //initlal text box count  
var FieldCount=1; //to keep track of text box added  
  
$(AddButton).click(function (e) //on add input button click  
{  
 if(x <= MaxInputs) //max input box allowed  
 {  
 FieldCount++; //text box added increment  
 //add input box  
 $(InputsWrapper).append('<div class="row"><div class="col-xs-4">Переменная</div><p class="col-xs-4"><input type="text" placeholder="Название переменной" class="form-control skill\_list" name="skill[]" id="field\_'+ FieldCount +'" value="Название переменной"/></p><a href="#" class="btn btn-danger removeclass">×</a></div>');  
 x++; //text box increment  
 }  
 return false;  
});  
  
$("body").on("click",".removeclass", function(e){ //user click on remove text  
 if( x > 1 ) {  
 $(this).parent('div').remove(); //remove text box  
 x--; //decrement textbox  
 }  
 return false;  
})  
  
 $('#submit').click(function(){  
 $.ajax({  
 url:"/postskill",  
 method:"POST",  
 data:$('#add\_skills').serialize(),  
 success:function(data)  
 { alert(data)  
 $('#resultbox').html(data);  
 $('#add\_skills')[0].reset();  
 }  
 });  
});  
  
});

</script>  
<style>  
.row {  
 padding:10px;  
}  
</style>  
<div class="container"><br /> <br />  
 <h2>{% block title %} Создать новый шаблон {% endblock %}</h2><div id="resultbox"></div>  
 {% if error %}  
 <div class="alert alert-danger" role="alert">{{ error }}</div>  
 {% endif %}  
 <hr>  
 <div class="form-group">  
 <form name="add\_skills" id="add\_skills" method="POST" enctype="multipart/form-data">  
 <div id="InputsWrapper">  
 <div class="row">  
 <div class="col-xs-4"><b>Название workflow</b></div>  
 <div class="col-xs-4"><input type="text" name="skill[]" placeholder="" class="form-control" /></div>  
 </div>  
 <div class="row">  
 <div class="col-xs-4">Переменная</div>  
 <div class="col-xs-4"><input type="text" name="skill[]" placeholder="Название переменной" class="form-control" /></div>  
 <div class="col-xs-4"><button type="button" name="add" id="AddMoreFileBox" class="btn btn-success">+</button></div>  
 </div>  
 </div>  
 <input type="file" name="file" class="col-xs-4">  
 <p>  
 <button type="submit" class="btn btn-primary  
 btn-sm" value="Upload">  
 Создать  
 </button>  
 </p>  
 </form>  
 </div>  
</div>  
</body>  
</html>  
{% endblock %}

На рисунке 1 представлена форма создания шаблона.

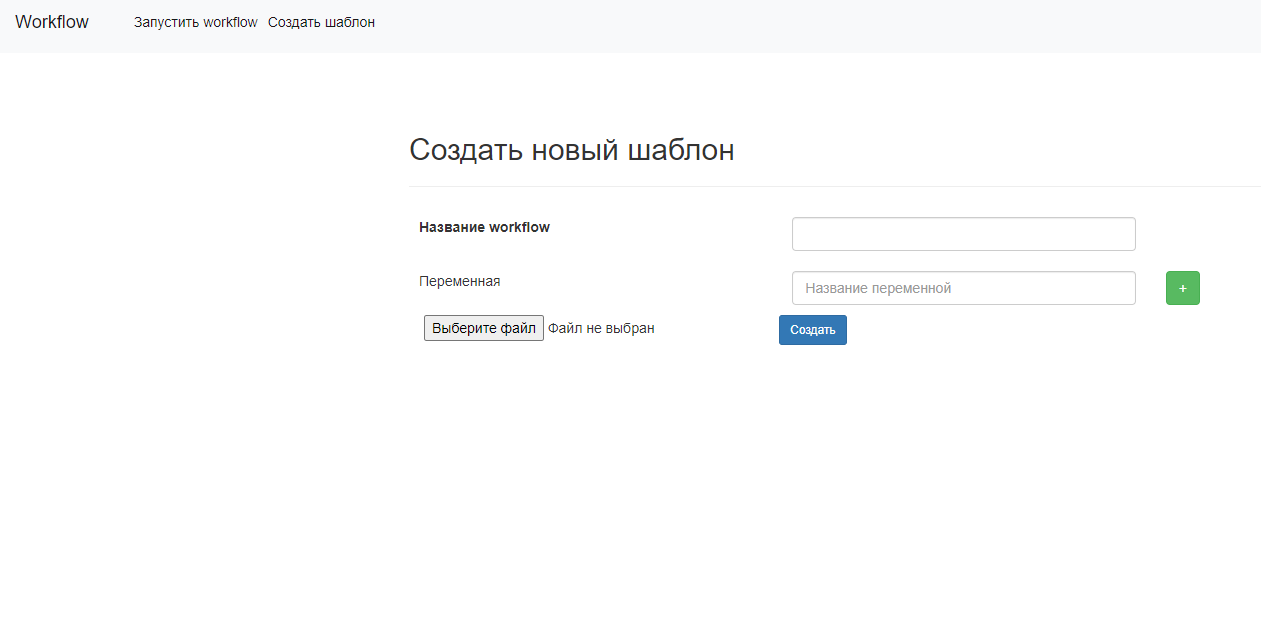


Рисунок 1 – Форма создания шаблона

На листинге 3 представлен код функции обработки запуска шаблона.

Листинг 3 – Код функции запуска шаблона

@app.route('/launch/<int:item\_id>', methods=["POST", "GET"])  
def launch(item\_id):  
 item = get\_item(item\_id)  
 if request.method == 'POST':  
 variables = {}  
 email = request.form.get("email")  
 vr = request.form.getlist("vars")  
 names = item["variables"]  
 for i in range(len(vr)):  
 variables[f"{{{{{names[i]}}}}}"] = vr[i]  
 filename = get\_filename\_by\_id(item\_id)[0]  
 tmp\_doc = Document(f"docs/{filename}")  
 result = "result.docx"  
 for k, v in variables.items():  
 for par in tmp\_doc.paragraphs:  
 replace\_text(par, k, v)  
 tmp\_doc.save(result)  
 send\_mail(email)  
 os.remove(result)  
 return redirect(url\_for('index'))  
 return render\_template('launch.html', item=item)

На листинге 4 представлена функция отправки письма.

Листинг 4 – Функция отправки письма

def send\_mail(email):  
 filepath = "result.docx"  
 basename = os.path.basename(filepath)  
 address = "\*"  
  
 # Compose attachment  
 part = MIMEBase('application', "octet-stream")  
 part.set\_payload(open(filepath, "rb").read())  
 encoders.encode\_base64(part)  
 part.add\_header('Content-Disposition', 'attachment; filename="%s"' % basename)  
  
 # Compose message  
 msg = MIMEMultipart()  
 msg['Subject'] = "Workflow project"  
 msg['From'] = address  
 msg['To'] = email  
 msg.attach(part)  
  
 # Send mail  
 server = smtplib.SMTP('smtp.gmail.com', 25)  
 server.starttls()  
 server.login(address, "\*")  
 text = msg.as\_string()  
 server.sendmail(address, email, msg.as\_string())  
 server.quit()

На рисунке 2 представлен внешний вид страницы отправки письма.

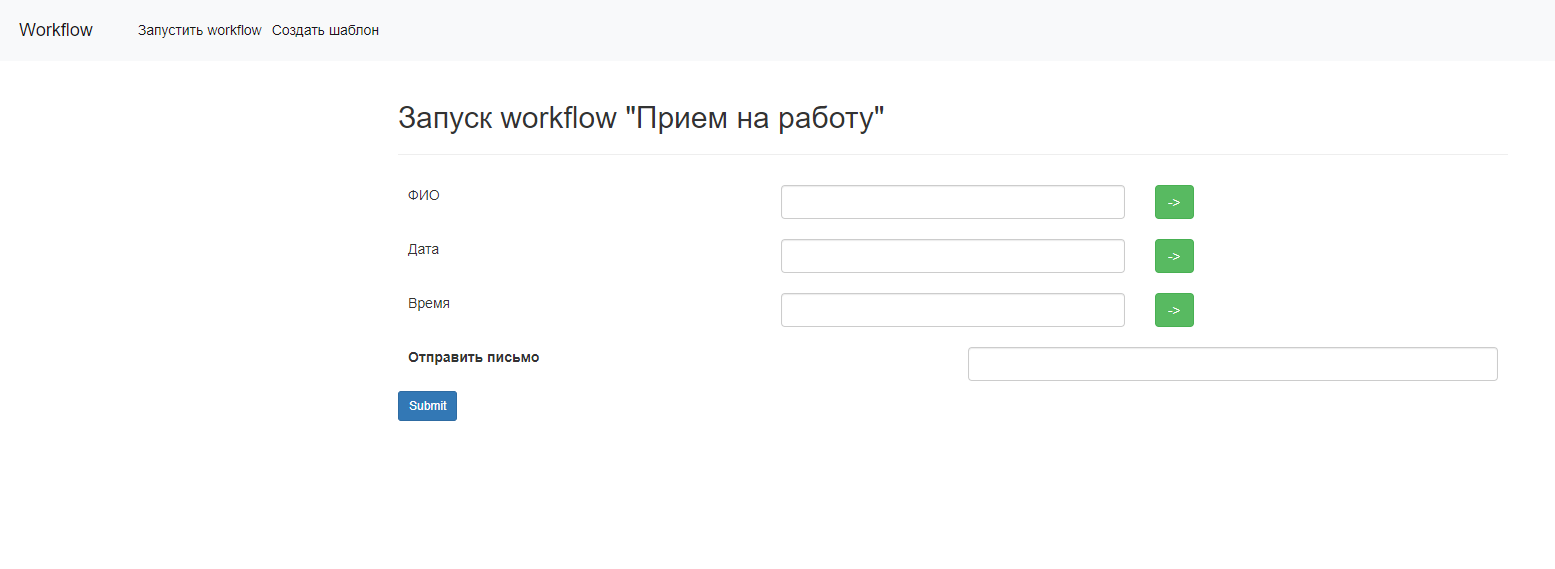


Рисунок 2 – Страница запуска workflow