



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по зачетной работе

Название: Зачётная работа

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-32Б
(Группа)

М.В.Щипицин
(И.О. Фамилия)

Преподаватель

Э.Р. Маняшев
(И.О. Фамилия)

Москва, 2023

Задание: Разработать собственное веб-приложение с авторизацией, аутентификацией, базой данных, локализацией.

Для разработки веб-приложения я использовал: python с библиотекой flask, PG Admin для работы с базой данных, JS для реализации функционала, html, CSS.

Проект представляет собой сайт для тренировки слепой печати.

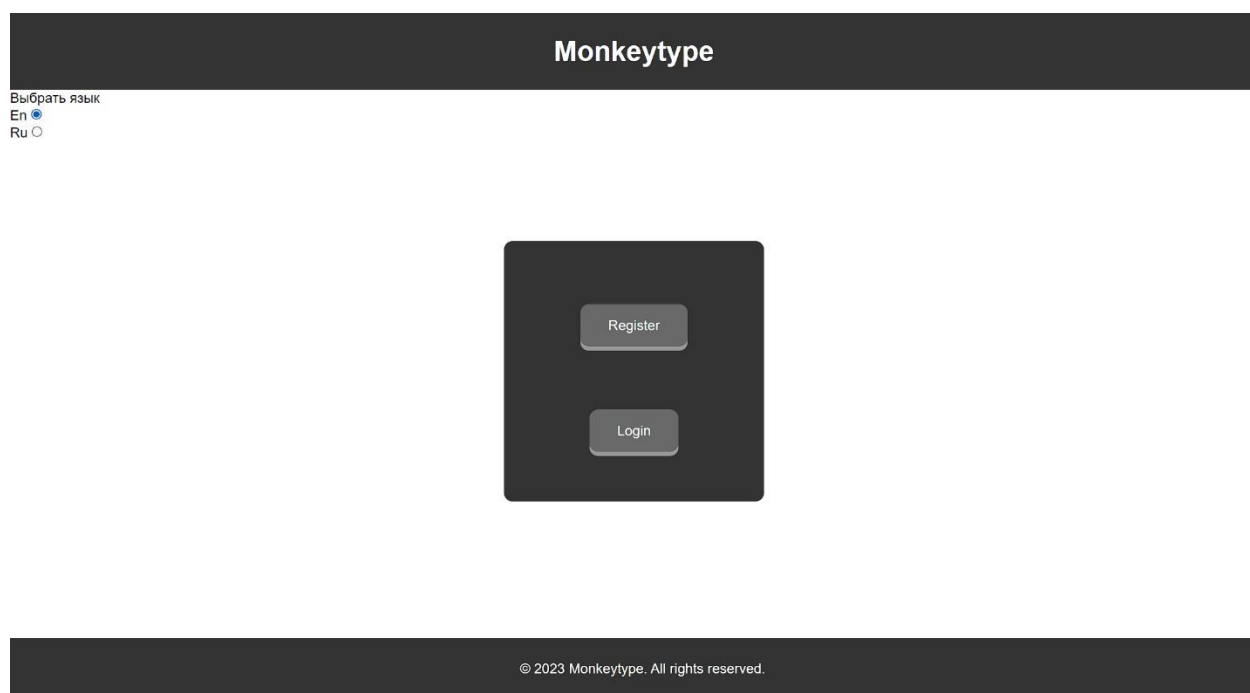


Рисунок 1 - интерфейс главной страницы

Для страниц регистрации и авторизации есть проверка на наличие пользователя в базе данных *Рисунок 2*.



Рисунок 2 – интерфейс страницы регистрации

На всех страницах в левом верхнем углу есть выбор языка на котором будет отображаться интерфейс всех страниц *Рисунок 1, Рисунок 2, Рисунок 3.*

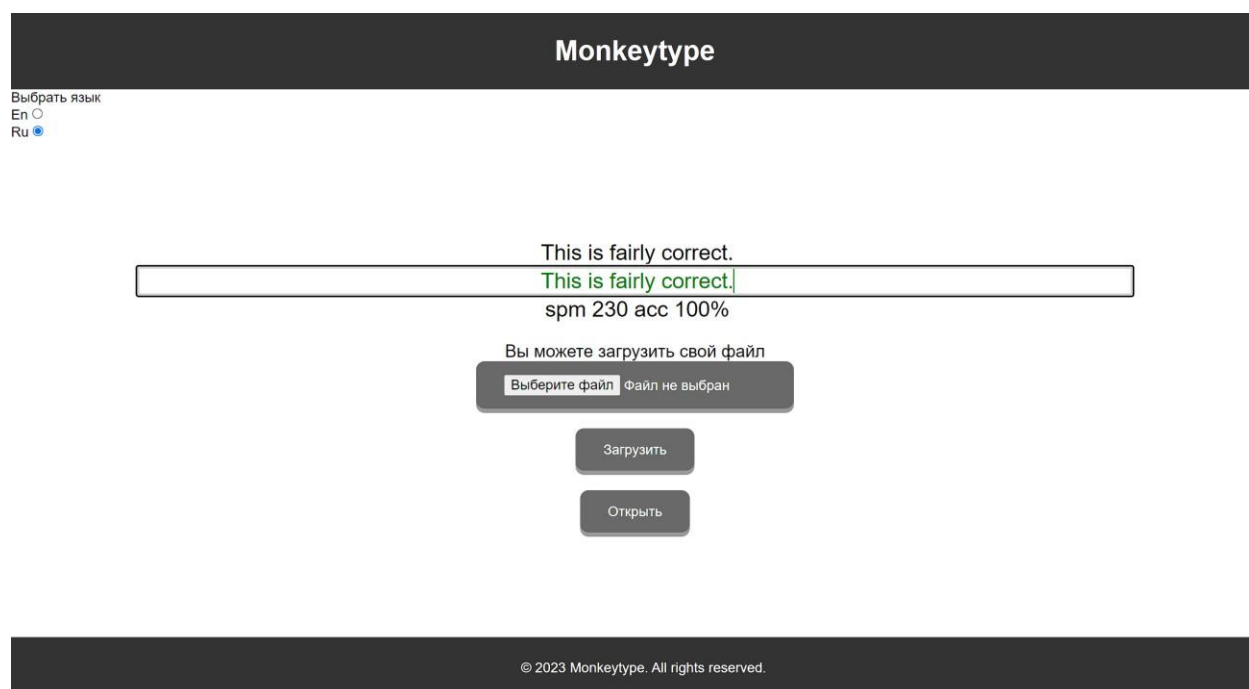


Рисунок 3 – интерфейс основной страницы

После записи определённой строки пользователем, выводится информация: *srm* – скорость печати, *acc* – точность печати. Пользователю доступна загрузка собственного файла для тренировки слепой печати на нём.

```

1  import unittest
2  from flask import Flask, render_template
3  import sys
4  sys.path.append('.')
5  from app import app
6
7  class TestFlaskApp(unittest.TestCase):
8      def setUp(self):
9          self.app = Flask(__name__)
10         self.app.config['TESTING'] = True
11         self.app.template_folder = '../templates'
12         self.client = app.test_client()
13
14     def test_home_page(self):
15         response = self.client.get('/')
16         self.assertEqual(response.status_code, 200)
17
18     def test_login_page(self):
19         response = self.client.get('/login')
20         self.assertEqual(response.status_code, 200)
21
22     def test_register_page(self):
23         response = self.client.get('/register')
24         self.assertEqual(response.status_code, 200)
25
26     def test_authorisation(self):
27         response = self.client.post('/login', data={'username': 'test', 'password': 'test'})
28         self.assertEqual(response.status_code, 302)
29
30 if __name__ == '__main__':
31     unittest.main()

```

Рисунок 4 – тесты для веб-приложения

Запрос

История запросов

1

select * FROM users;

Data Output

Сообщения

Notifications

≡+

	id [PK] integer	username character varying (80)	email character varying (120)	password character varying (120)	file_pols text
1	2	abobaasd	maksgycas1@gmail.com	aboba	[null]
2	3	amogus	amogus@gmail.com	aboba	[null]
3	1	aboba	maksgyc1@gmail.com	aboba	never.txt
4	4	test	test@test.com	test	[null]
5	5	maks	maks@mail.com	maks	never.txt

Рисунок 5 – таблица из базы данных веб-приложения

Все данные внесённые на сайте будут записаны в базу данных. Это данные пользователей: логин, пароль, почта, имя файла.

Рисунок 5.

Вывод: При выполнении данной работы я научился создавать свои вебприложения с регистрацией, авторизацией, базой данных. Также я научился тестировать веб-приложения с помощью библиотеки flask для python

Рисунок 4.