



SHMYAKS

# 01

## Сервис

### Классификация фото-сканов

Проблема заключается в необходимости автоматизировать и упростить процесс анализа и классификации фото-сканов документов с определенной вероятностью, а также извлечения основной информации из каждого документа.

# 01

## Автоматизация

### Автоматизация процесса

Сокращение времени, затрачиваемого на заполнение анкет и проверку документов.

# 02

## Accuracy

### Точность и эффективность

Использование алгоритма распознавания Yolo-8 обеспечивает высокую точность распознавания документов.

# 03

## Интеграция

### Простота внедрения

Веб-приложение готово к интеграции на основной сайт компании, что обеспечивает легкость внедрения и использования.



```
import numpy as np
import cv2
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

def classify(cv2_img, model, launch_type):
    img_array = np.expand_dims(cv2.resize(cv2_img, (224, 224)),
axis=0).astype('float32') / 255.

    predictions = model.predict(img_array)[0]

    predictions_sum = np.sum(predictions)
    normalized_predictions = predictions / predictions_sum

    class_indices = {'pass_1': 0, 'pass_2': 1, 'pts': 2, 'sts_1': 3, 'sts_2': 4,
'vu_1': 5, 'vu_2': 6}
    classes = list(class_indices.keys())

    predicted_class = classes[np.argmax(predictions)]

    logger.info('[classifier] got image type predictions')
    percentages = {classes[i]: round(float(normalized_predictions[i]) * 100, 2) for
i in range(len(predictions))}

    return predicted_class, percentages
```



## ML составляющая

Подготовка картинки: Сначала мы делаем картинку правильного размера, чтобы её легко было посмотреть.

Определение: Затем мы показываем картинку нашему волшебному помощнику.

Ответ: Волшебный помощник смотрит и говорит нам, что он видит на картинке.

Результаты: В итоге мы узнаем, что на картинке, и сколько уверенности у волшебного помощника в своём ответе.



```
@router.post("/detect")
async def detect_image(data: ImageData):
    try:
        image_bytes = base64.b64decode(data.image)
    except Exception as e:
        raise HTTPException(status_code=400, detail="Base64 decoding error") from e

    unique_filename = f"{uuid.uuid4()}.png"

    os.makedirs('uploaded_images', exist_ok=True)
    file_path = os.path.join('uploaded_images', unique_filename)

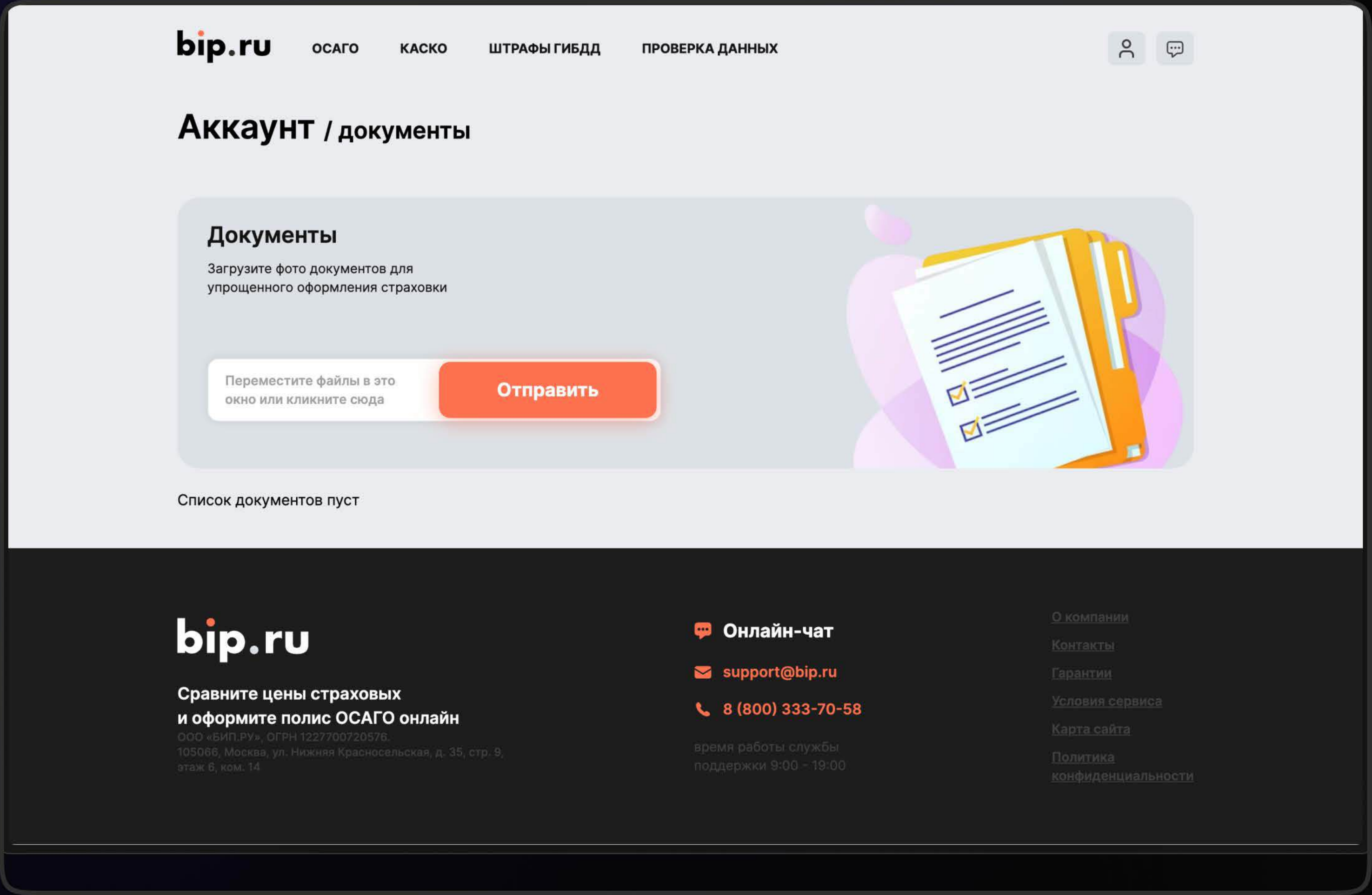
    with open(file_path, 'wb') as image_file:
        image_file.write(image_bytes)

    result = core.main(
        doc=file_path,
        normalizer_model=normalizer_model,
        classifier_model=classifier_model,
        text_model=text_model,
        debugging=False,
        reader=reader,
        launch_type="linux"
    )
```

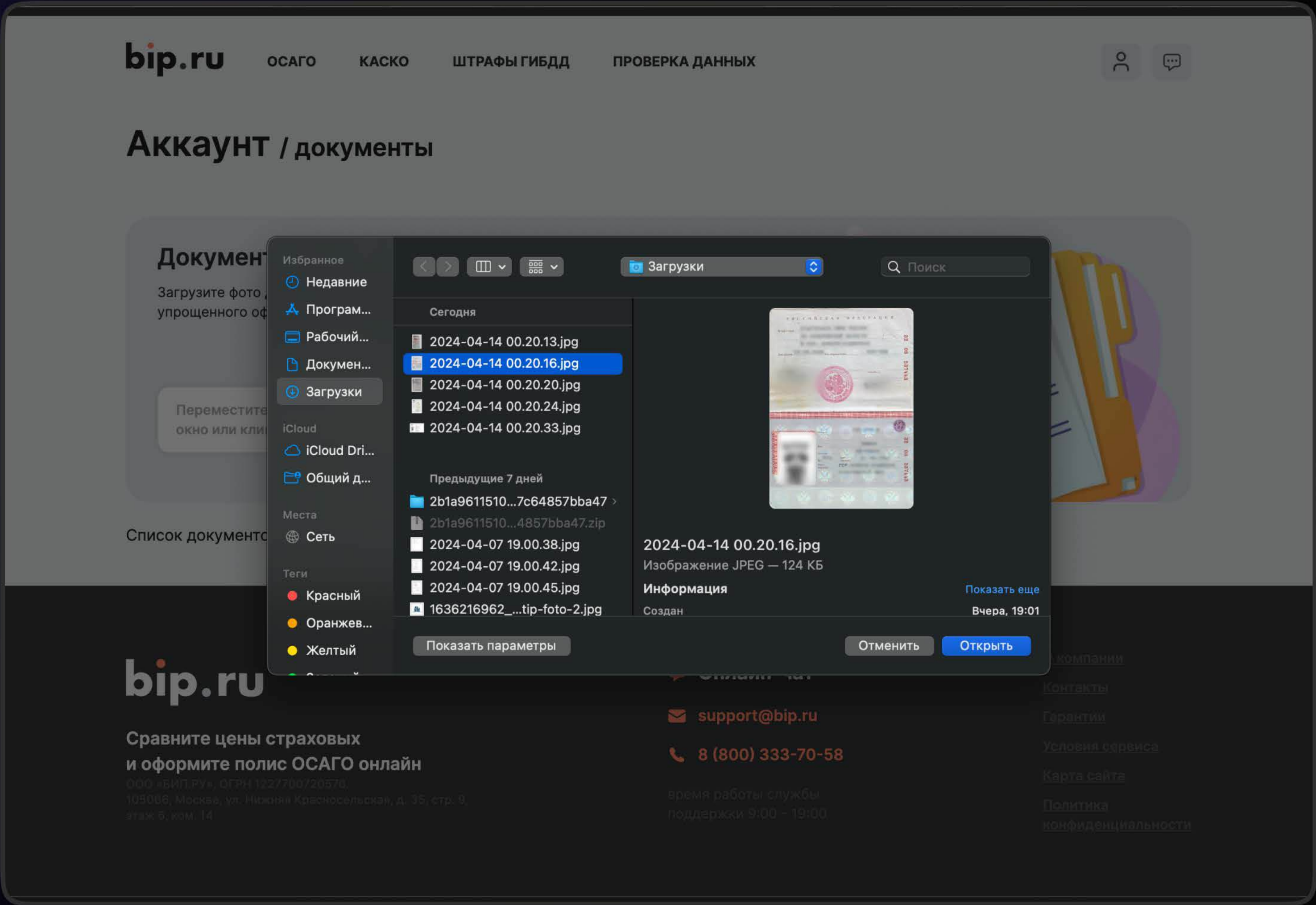


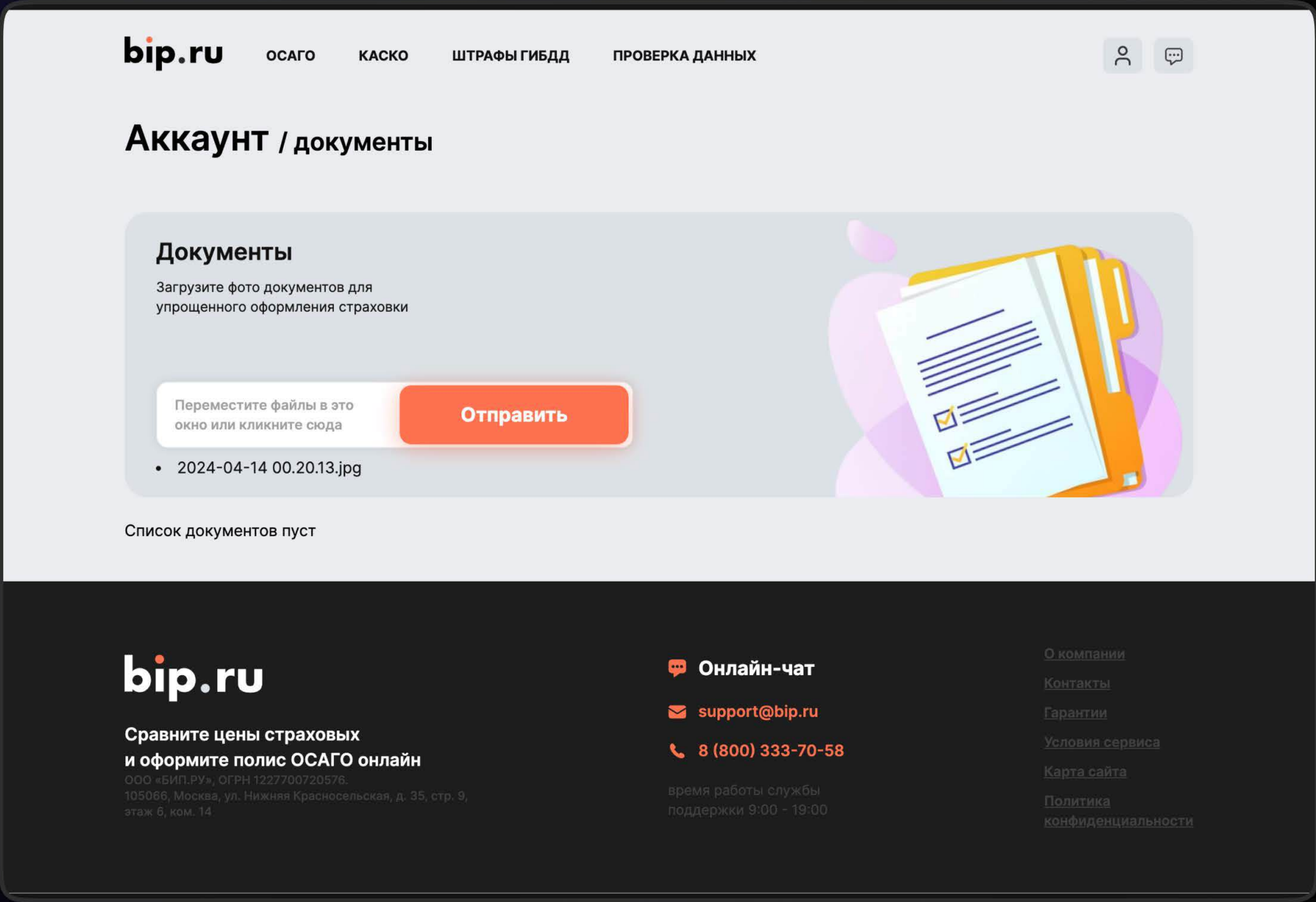
## Backend составляющая

Бекенд сайта реализован на golang и fastapi. На go обеспечивается взаимодействие с историей распознаваний, на fastapi взаимодействие с моделью. Разделение по языкам обусловлено удобством работы с моделью через python, тк сама модель написана на нем. В Go части был использован gin фреймворк и postgresql база. В FastAPI используется middleware для ограничения количества запросов от юзеров для защиты сервера от перегрузок.











### Документы

Загрузите фото документов для  
упрощенного оформления страховки

Переместите файлы в это  
окно или кликните сюда

Отправить



### Паспорт стр. 1



Открыть

**bip.ru**

Сравните цены страховых  
и оформите полис ОСАГО онлайн

ООО «БИП.РУ», ОГРН 1227700720576,  
105066, Москва, ул. Нижняя Красносельская, д. 35, стр. 9,  
этаж 6, ком. 14

 **Онлайн-чат**

 **support@bip.ru**

 **8 (800) 333-70-58**

время работы службы  
поддержки 9:00 - 19:00

[О компании](#)

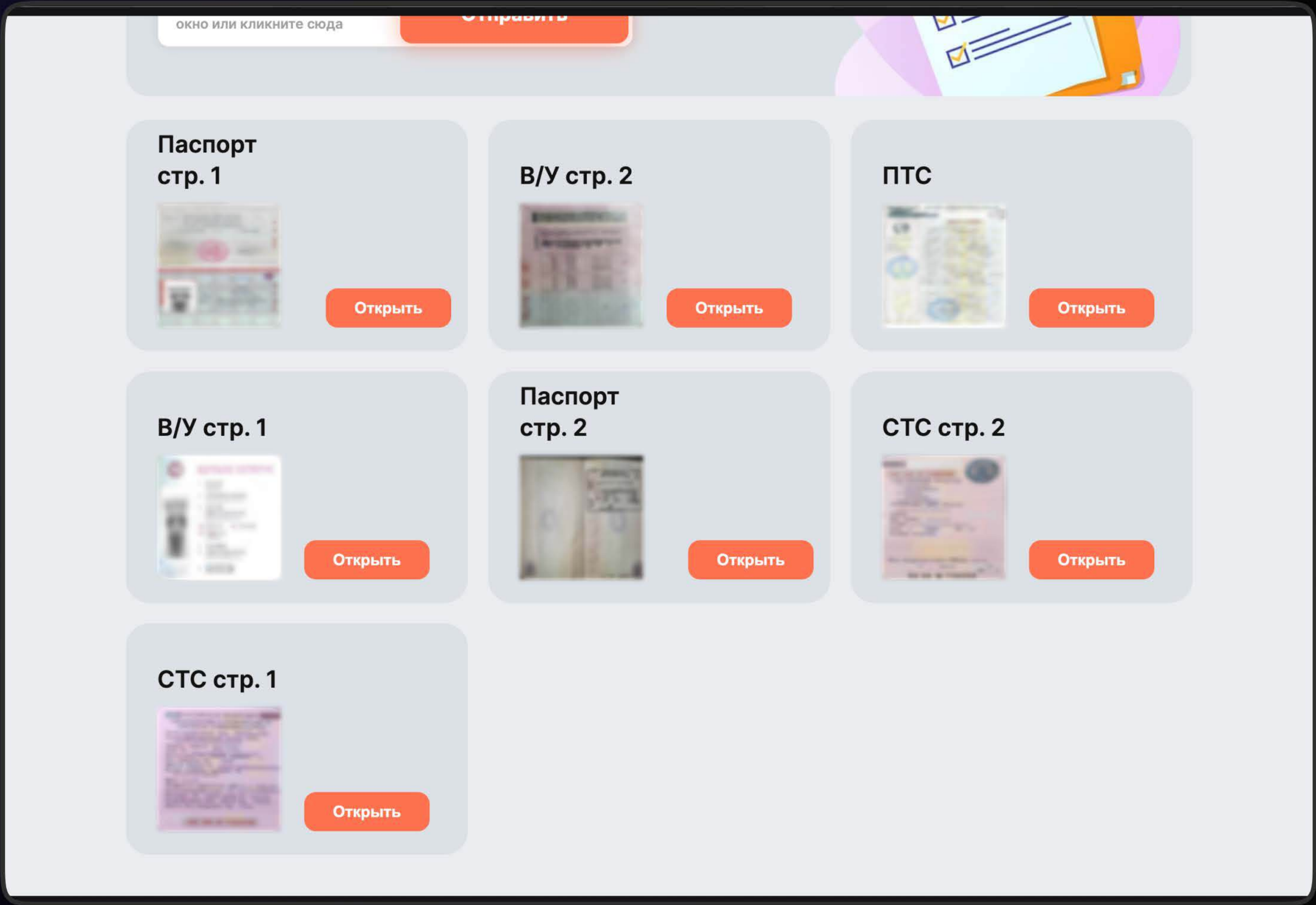
[Контакты](#)

[Гарантии](#)

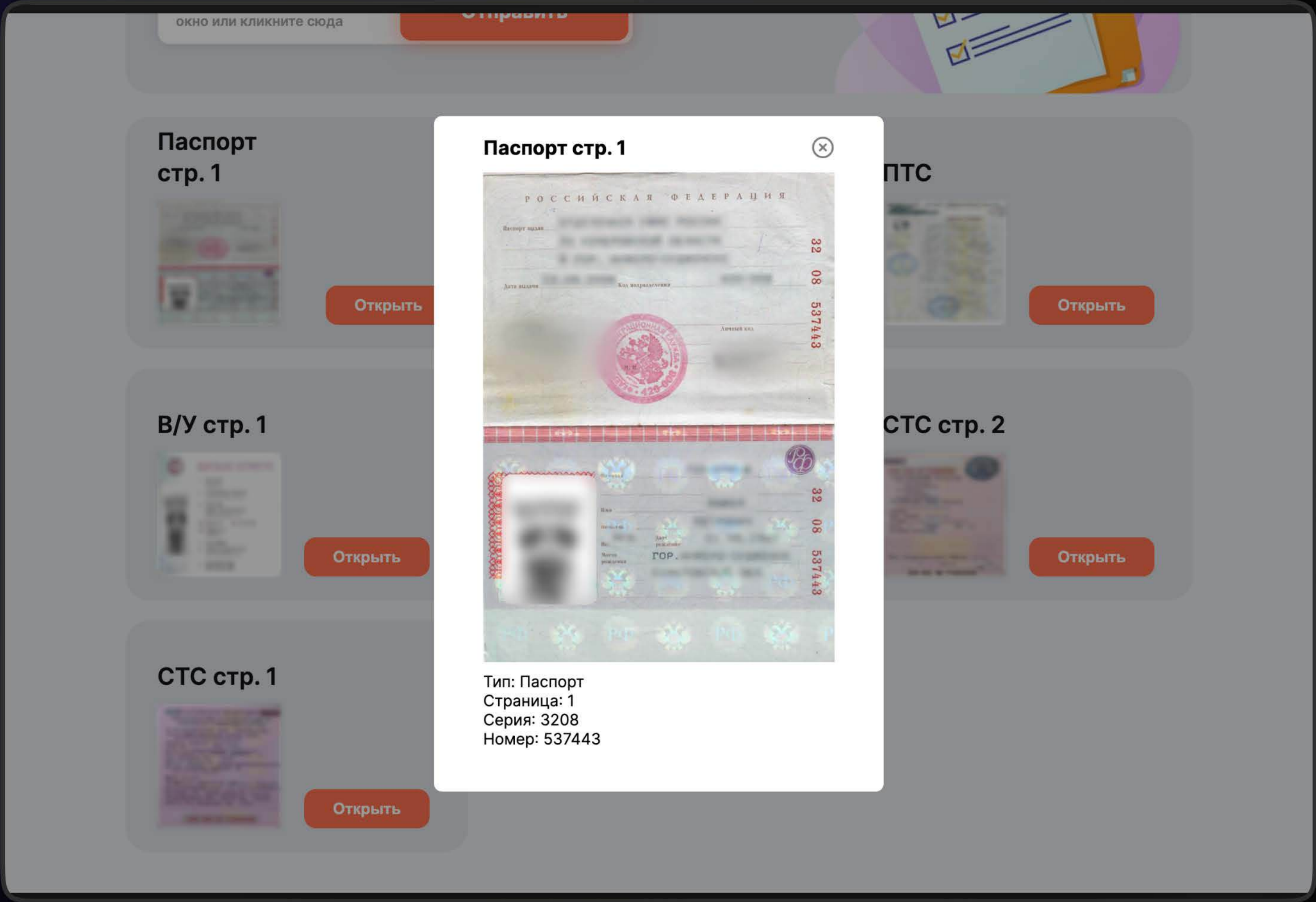
[Условия сервиса](#)

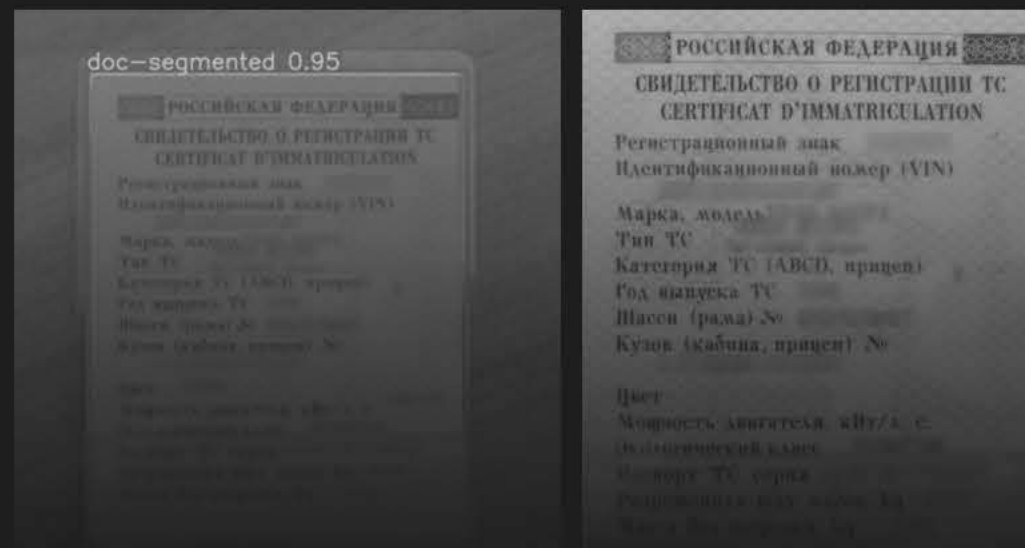
[Карта сайта](#)

[Политика  
конфиденциальности](#)









**Грубая деформация**  
Выравнивает фото и сканы

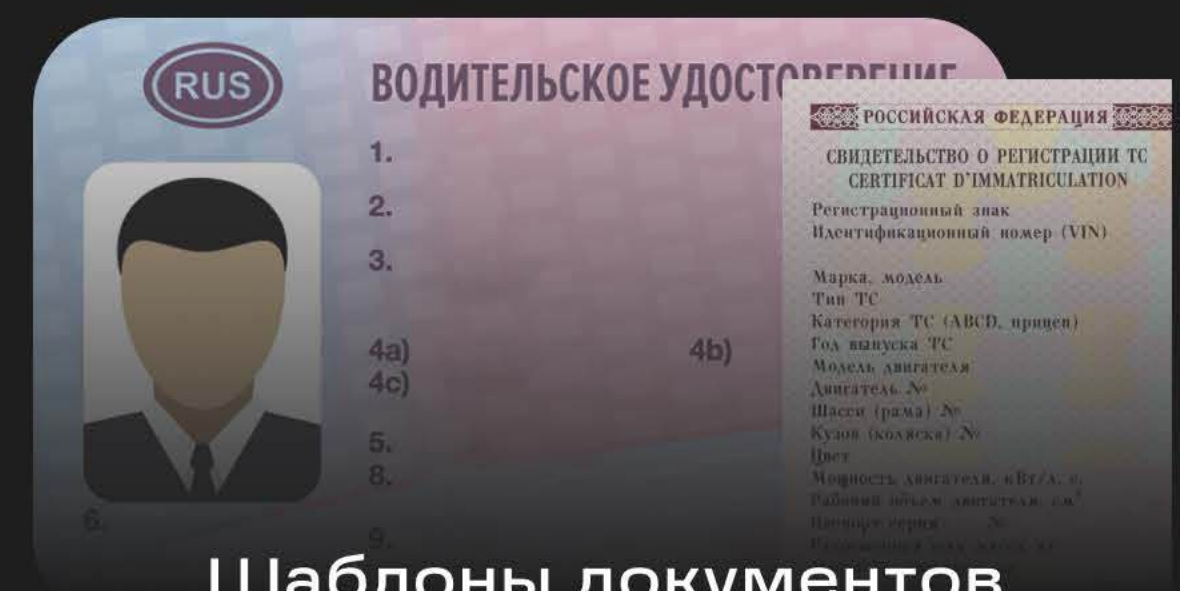


Паспорт 96%

ПТС 2%

СТС 1%

В/У 1%



**Шаблоны документов**  
для автоматического заполнения данных



**HARD SKILLS**  
более 15 технологий используется в проекте



**SHMYAKS**

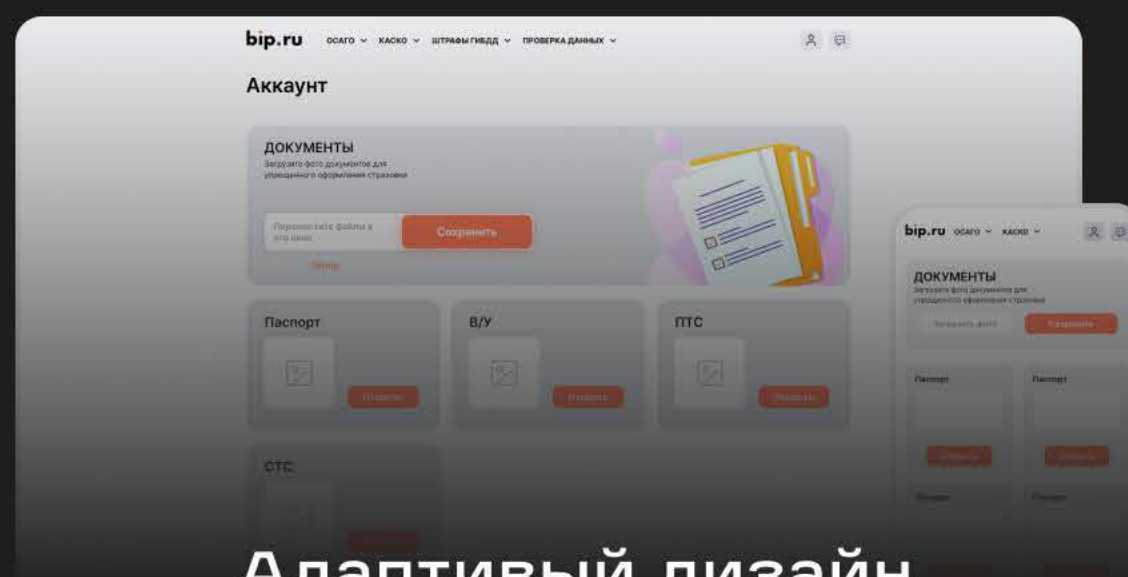
**3 фото**  
в секунду

Подтвердите действие на [gagarin.shmyaks.ru](https://gagarin.shmyaks.ru)

Слишком много запросов, пожалуйста, попробуйте позже

OK

**Middleware**  
для предотвращения DDOS-атак



**Адаптивный дизайн**  
с возможностью интеграции на bip.ru



**Метрика**  
результаты тренировки модели классификации

**Yolo v8**



## 01 Расширение

### Подключение к базе ГАИ

Позволит дополнительно проверять данные пользователей и подтверждать их

## 02 Интеграция

### Кладр адресов и данных


Дополнительное улучшение работы нашей модели, обеспечение более точные результаты

## 03 Функционал

### Удобство и практичность

В дальнейшем должно быть распознавание всех данных, а также реализована полноценная работа с документами в личном кабинете





NotionMiroFigJam

Project Manager

Глеб Менеджеров

@crowwwdy




FigmaPhotoshop

UX/UI

Артем Фигмов

@dewerrr



SCSSReduxReact

Front-End

Константин Фронтендов

@kokoan23




GOSQLDocker

Back-End

Михаил Девопсов

@kelianis



YOLOTensorFlow

ML

Иван Сивишкин

@cocucku13





СПАСИБО!