


```
```env
SECRET_KEY=замените-на-случайную-строку
SMTP_SERVER=smtp.yandex.ru
SMTP_PORT=587
SMTP_USER=your-email@yandex.ru
SMTP_PASSWORD=your-app-password
```
```

> ****Важно:**** для Яндекс Почты используйте пароль приложения, а не основной пароль аккаунта.
> Создать пароль приложения: [id.yandex.ru → Безопасность → Пароли приложений] (<https://id.yandex.ru/security/app-passwords>)

4. Запустить приложение

```
```bash
python app.py
```
```

Приложение будет доступно по адресу:
[<http://localhost:5000>] (<http://localhost:5000>)

Деплой в Yandex Cloud Functions

Требования

- Аккаунт в [Yandex Cloud] (<https://cloud.yandex.ru/>)
- Установленный [Yandex Cloud CLI (`yc`)] (<https://cloud.yandex.ru/docs/cli/quickstart>)

1. Создать ZIP-архив для деплоя

Архив должен содержать все файлы ****в корне**** (без вложенной папки):


```
```bash
Windows (PowerShell)
Compress-Archive -Path app.py, handler.py, requirements.txt, collages,
static -DestinationPath deploy.zip -Force

Linux / macOS
zip -r deploy.zip app.py handler.py requirements.txt collages/ static/
```
```

2. Создать или обновить функцию через консоль Yandex Cloud

1. Перейдите в [Yandex Cloud Console → Cloud Functions] (<https://console.cloud.yandex.ru/functions>)
2. Создайте новую функцию (или откройте существующую)
3. Создайте новую версию:
 - ****Среда выполнения:**** `python312`
 - ****Способ загрузки:**** ZIP-архив → загрузите `deploy.zip`
 - ****Точка входа:**** `handler.handler`
 - ****Память:**** 256 МБ
 - ****Таймаут:**** 30 с

3. Задать переменные окружения функции

>  ****Важно:**** файл `.env` не читается в Cloud Functions. Переменные нужно задавать напрямую в настройках функции в консоли Yandex Cloud.

> Без этого функция вернёт ошибку ``"SMTP не настроен: задайте переменные окружения SMTP_USER и SMTP_PASSWORD"`.`

Как задать переменные окружения в Yandex Cloud Console:

1. Перейдите в **Cloud Functions** → выберите функцию
2. Нажмите **Редактировать** (или создайте новую версию)
3. Прокрутите до раздела **Переменные окружения**
4. Добавьте следующие переменные:

| Ключ | Значение | Описание |
|------------------------------|-------------------------------------|--|
| <code>`SECRET_KEY`</code> | случайная строка | Секретный ключ Flask-сессий |
| <code>`SMTP_SERVER`</code> | <code>`smtp.yandex.ru`</code> | SMTP-сервер Яндекс Почты |
| <code>`SMTP_PORT`</code> | <code>`587`</code> | Порт SMTP (STARTTLS) |
| <code>`SMTP_USER`</code> | <code>`your-email@yandex.ru`</code> | Email-адрес отправителя |
| <code>`SMTP_PASSWORD`</code> | пароль приложения Яндекс | Пароль приложения (не основной пароль) |

5. Сохраните — будет создана новая версия функции

> **Пароль приложения Яндекс:** [id.yandex.ru → Безопасность → Пароли приложений] (<https://id.yandex.ru/security/app-passwords>)

4. Создать API Gateway

Чтобы функция была доступна по HTTP:

1. Перейдите в **API Gateway** → Создать шлюз
2. Используйте спецификацию:

```
```yaml
openapi: 3.0.0
info:
 title: congratulate-your-teacher
 version: 1.0.0
paths:
 /:
 x-yc-apigateway-any-method:
 x-yc-apigateway-integration:
 type: cloud_functions
 function_id: <ID вашей функции>
 service_account_id: <ID сервисного аккаунта>

 /preview:
 x-yc-apigateway-any-method:
 x-yc-apigateway-integration:
 type: cloud_functions
 function_id: <ID вашей функции>
 service_account_id: <ID сервисного аккаунта>

 /send:
 x-yc-apigateway-any-method:
 x-yc-apigateway-integration:
 type: cloud_functions
```

```
function_id: <ID вашей функции>
service_account_id: <ID сервисного аккаунта>
...
```

---

### ## Как работает приложение

1. **Шаг 1** – пользователь загружает фото ученика и учителя, вводит текст поздравления (или оставляет пустым – выберется случайная фраза)
2. **"/preview"** – сервер создаёт коллаж, накладывая фото на случайный шаблон из `collages/`, и возвращает его в base64 для предпросмотра
3. **Шаг 2** – пользователь вводит email и нажимает «Отправить»
4. **"/send"** – сервер отправляет его письмом через SMTP

---

### ## Зависимости

Пакет	Версия	Назначение
Flask	2.3.3	Веб-фреймворк
Pillow	10.0.1	Создание и обработка изображений
Werkzeug	2.3.7	Утилиты Flask (загрузка файлов)