Программа для ЭВМ «Обучай»

Руководство системного программиста

Листов 18

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

2026

АННОТАЦИЯ

Документ содержит описание структуры, процедуры установки, настройки, проверки, обновления и резервного копирования компонентов программы для ЭВМ «Обучай» и предназначен для системных программистов, осуществляющих подготовку Программы к эксплуатации.

Настоящий документ предназначен для персонала, имеющего квалификацию системного администратора и навыки администрирования ОС семейства Linux (Debian, Ubuntu), среды контейнеризации Docker и Docker Compose, СУБД PostgreSQL, объектного хранилища MinIO и веб-сервера Caddy.

Документ не является руководством пользователя по установке и настройке, операционной системы, и прочего системного и технологического программного обеспечения.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ....4
1.1 Серверная часть....4
1.2 Микросервис сборки презентаций....4
1.3 Клиентская часть....4
1.4 СУБД, хранилище и веб-сервер....4
2. СТРУКТУРА ПРОГРАММЫ....6
2.1 Состав контейнеров....6
2.2 Основные модули серверной части....6
2.3 Взаимодействие компонентов....7
2.4 База данных и файлы....7
3. ТЕХНИЧЕСКИЕ СРЕДСТВА....8
3.1 Проверочный VPS....8
3.2 Компьютер эксперта....8
4. КОМПЛЕКТ ПОСТАВКИ И УСТАНОВКА....9
4.1 Каталоги....9
4.2 Подготовленный удалённый экземпляр....9
4.3 Автоматическая установка из комплекта....10
5. ПАРАМЕТРЫ КОНФИГУРАЦИИ....11
6. ПРОВЕРКА РАБОТОСПОСОБНОСТИ....12
6.1 Проверка на сервере....12
6.2 Независимая проверка с компьютера эксперта....12
6.3 Проверка ИИ....12
7. ОБНОВЛЕНИЕ И ЖУРНАЛЫ....13
8. РЕЗЕРВНОЕ КОПИРОВАНИЕ....14
9. ДОСТУП И ЛИЦЕНЗИОННЫЕ МЕХАНИЗМЫ....15
10. ТИПОВЫЕ НЕИСПРАВНОСТИ....16

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Программа для ЭВМ «Обучай» (далее — Программа) — веб-сервис для автоматизированной подготовки педагогом учебных материалов и организации работы с учениками. Проверочный экземпляр работает на сервере с ОС Ubuntu 26.04 LTS, архитектура x86-64. На VPS под управлением Docker Compose работают пять контейнеров приложения и шестой публичный Caddy с лендингом, веб-шлюзом и обратным прокси к приложению и API. В VM без публичного профиля запускаются пять контейнеров приложения.

1.1 Серверная часть

Программный сервис на Python 3.12, FastAPI и Uvicorn. Реализует REST API, бизнес-логику, конвейеры формирования материалов, печатные документы и фоновые задачи. Образ obuchai-backend:rf-expertise содержит исходные модули, зависимости, LibreOffice, средства формирования документов и локальные шрифты.

1.2 Микросервис сборки презентаций

Программный сервис на Kotlin 2.0, Ktor и Apache POI. Образ obuchai-presentations:rf-expertise содержит исполняемый JAR и JVM 17. Сервис формирует PPTX и обрабатывает шаблоны презентаций.

1.3 Клиентская часть

Одностраничное приложение на TypeScript, React 18 и Vite. Результат сборки — статические файлы в /opt/obuchai-rf/deploy/front, включая shell.html и предварительно сформированные публичные страницы. Шрифты раздаются локально.

1.4 СУБД, хранилище и веб-сервер

- PostgreSQL 16 — образ postgres:16;

- MinIO-совместимое S3-хранилище — образ pgsty/silo:latest;
- Caddy 2 — образ caddy:2, отдача клиентской части и маршрутизация /api/*.

2. СТРУКТУРА ПРОГРАММЫ

2.1 Состав контейнеров

Таблица 1 – Состав контейнеров проверочного экземпляра

Контейнер	Образ	Порт	Назначение
obuchai-rf-caddy	caddy:2	18080 на loopback	приложение и обратный прокси
obuchai-rf-public-site	caddy:2	80, 443 на VPS	HTTPS-лендинг, веб-шлюз и proxy к приложению/API через внутренний Caddy
obuchai-rf-backend	obuchai-backend:rf-expertise	8000 внутри Docker	API и бизнес-логика
obuchai-rf-presentations	obuchai-presentations:rf-expertise	8050 внутри Docker	формирование PPTX
obuchai-rf-postgres	postgres:16	5432 внутри Docker	база данных
obuchai-rf-minio	pgsty/silo:latest	9000, 9001 внутри Docker	файловое хранилище

PostgreSQL, MinIO, backend и микросервис презентаций не публикуются наружу. Caddy приложения принимает соединения на 127.0.0.1:18080 и через внутреннюю сеть public_app. UFW разрешает SSH 22/tcp и порты 80/443 публичного Caddy. Публичный Caddy не подключён к сети backend и проксирует запросы только на внутренний Caddy. Пользователь открывает приложение по HTTPS через веб-шлюз; SSH-туннель сохранён как дополнительный доступ.

2.2 Основные модули серверной части

Исходный текст находится в /opt/obuchai-rf/source/backend/app. Основные пакеты: auth и account_deletion; worksheets, tasks и worksheet_design; materials, lessons, documents и cards; presentations; interactives; exams, pedagogy и curriculum; classroom, assignments, submissions и checks; library, sources и storage_cleanup; bank; payments и offers; llm_routing и llm_gateway; requests.

Точка входа — app/main.py. Команда запуска внутри контейнера: uvicorn app.main:app --host 0.0.0.0 --port 8000.

2.3 Взаимодействие компонентов

Браузер обращается к публичному Caddy по HTTPS либо к внутреннему Caddy через локальный SSH-туннель. Публичный Caddy отдаёт лендинг и проксирует приложение и API на внутренний Caddy. Страницы защищены Basic Auth, API использует Bearer-токены, самостоятельная регистрация возвращает 403. Backend использует PostgreSQL через пул соединений, MinIO по протоколу S3 и микросервис презентаций по внутреннему HTTP с проверкой служебного ключа. Внешние вызовы ИИ выполняются только серверной частью через настроенный шлюз RouterAI.

2.4 База данных и файлы

Схема формируется из source/backend/db_init.sql и SQL-миграций в source/backend/migrations. Данные PostgreSQL хранятся в volume obuchai-rf_pgdata. Файлы материалов, изображения, презентации и превью хранятся в volume obuchai-rf_miniodata.

Поставка содержит три отдельные тестовые учётные записи, 10 редакционных тем презентаций и демонстрационную презентацию на 20 слайдов. Производственная база и данные реальных пользователей не копируются.

3. ТЕХНИЧЕСКИЕ СРЕДСТВА

3.1 Проверочный VPS

Фактическая конфигурация подготовленного стенда: 2 vCPU x86-64, 2 ГБ ОЗУ, 2 ГБ swap, 40 ГБ диска, Ubuntu 26.04 LTS. Для повторной установки и расширенной нагрузки рекомендуется 4 vCPU, 8 ГБ ОЗУ и 60 ГБ диска.

3.2 Компьютер эксперта

Компьютер с браузером актуальной версии, Python 3 и OpenSSH. Дополнительные библиотеки для автоматической проверки scripts/remote_check.py не требуются.

4. КОМПЛЕКТ ПОСТАВКИ И УСТАНОВКА

4.1 Каталоги

Корневая директория: /opt/obuchai-rf. Основные компоненты:

- source/backend, source/frontend, source/presentations — исходный текст трёх компонентов;
- deploy/docker-compose.yml, deploy/Caddyfile и deploy/Caddyfile.public — конфигурация запуска;
- deploy/.env — секретная конфигурация экземпляра, права 600;
- deploy/front — собранная клиентская часть;
- deploy/public-site — лендинг и публичные документы; публичный Caddy проксирует приложение и API только через внутренний Caddy по сети public_app;
- dist/images.tar.gz — пять container images;
- scripts — установка, запуск, проверка, резервное копирование;
- docs — документация и журнал проверки;
- SOURCE_MANIFEST.json — идентификаторы версий и образов.

4.2 Подготовленный удалённый экземпляр

Для эксперта подготавливается уже установленный VPS. Повторная сборка и обращения к Docker Hub, npm, pip, GitHub или Google Fonts не нужны. Отдельные приватный SSH-ключ, credentials.txt и закреплённый ssh_known_hosts передаются по защищённому каналу.

Приложение открывается по адресу https://obuchai-public.ru/generator. Сначала выполняется вход в веб-шлюз с логином expert, затем вход в тестовую учётную запись; пароли приведены только в приватном credentials.txt. API использует Bearer-токены, самостоятельная регистрация заблокирована. Полные реквизиты сервера приведены в приватной инструкции docs/EXPERT.md. Дополнительно можно запустить SSH-туннель с локального порта 8080 на 127.0.0.1:18080 и открыть http://localhost:8080.

4.3 Автоматическая установка из комплекта

При необходимости повторного развёртывания на подготовленной ОС с Docker выполняется /opt/obuchai-rf/scripts/install.sh. Сценарий:

- проверяет deploy/.env и запрещает запуск с незаполненными обязательными параметрами;
- загружает пять образов из локального dist/images.tar.gz;
- запускает PostgreSQL и MinIO и ожидает их готовности;
- создаёт базу при первичной установке, применяет db_init.sql и миграции с ON_ERROR_STOP=1;
- запускает backend, presentations и Caddy и ожидает healthcheck backend;
- создаёт тестовые учётные записи, 10 тем и демонстрационную презентацию;
- выполняет scripts/smoke.sh.

Повторный запуск полной первичной инициализации на эксплуатируемой базе не выполняется. На VM первичный запуск контролируется systemd unit obuchai-rf и меткой /var/lib/obuchai-rf/installed.

5. ПАРАМЕТРЫ КОНФИГУРАЦИИ

Таблица 2 – Основные параметры проверочного экземпляра

Параметр	Назначение
APP_ENV=stand	изолированный проверочный контур
PUBLIC_URL, BACKEND_URL, FRONTEND_URL, FRONT_ORIGIN	адреса экземпляра
APP_BIND_ADDRESS=127.0.0.1, APP_HTTP_PORT=18080	внутренний Caddy на loopback; HTTPS через публичный Caddy и дополнительный SSH-туннель
COMPOSE_PROFILES=public-site	включение отдельного публичного контейнера на VPS
PUBLIC_SITE_ADDRESS, PUBLIC_BIND_ADDRESS	домен публичного стенда и адрес внешних портов 80/443
DB_HOST, DB_PORT, DB_NAME, DB_USER, DB_PASSWORD	подключение к PostgreSQL
MINIO_ENDPOINT, MINIO_ACCESS_KEY, MINIO_SECRET_KEY	подключение к S3-хранилищу
JWT_SECRET	подпись токенов тестовых сессий
INTERNAL_API_KEY	внутренний обмен с микросервисом презентаций
PRESENTATIONS_SERVICE_URL	адрес presentations:8050 внутри Docker
LLM_PROVIDER=routerai	выбор российского шлюза моделей
ROUTERAI_URL, ROUTERAI_API_KEY	адрес и ключ RouterAI
LLM_ROUTING_FILE	файл фактических маршрутов и моделей
CLOUDPAYMENTS_PUBLIC_ID/ API_SECRET	пустые значения: платежи отключены
EMAIL_TEST_ALLOWLIST	несуществующий адрес .invalid: исходящая почта отключена
DISABLE_MAX_WEBHOOK=1	webhook и бот отключены
INDEXNOW_DISABLED=1	обращения к IndexNow отключены

Платёжные операции, выплаты авторам, исходящая почта, OAuth, Sentry, LangSmith, Microsoft Clarity и сторонняя аналитика в проверочном профиле отключены. Браузерные сторонние подключения дополнительно запрещены Content-Security-Policy.

6. ПРОВЕРКА РАБОТОСПОСОБНОСТИ

6.1 Проверка на сервере

```
cd /opt/obuchai-rf/deploy && docker compose ps
/opt/obuchai-rf/scripts/smoke.sh
```

Smoke-тест проверяет frontend, ответ API 401 без токена, ответ API 200 с реальным запросом к PostgreSQL, готовность PostgreSQL и наличие не менее 10 тем презентаций.

6.2 Независимая проверка с компьютера эксперта

Из каталога выданного комплекта выполняется python3 scripts/remote_check.py. Сценарий проверяет HTTPS-лендинг, веб-шлюз, Bearer API, запрет самостоятельной регистрации и ответы 404 для служебных путей. Затем он входит в три тестовых аккаунта через домен и SSH-туннель, проверяет /auth/me, 10 тем, демонстрацию на 20 слайдов и реальные выгрузки PPTX и PDF. Туннель создаётся и закрывается автоматически, пароли и ключи в вывод не включаются.

6.3 Проверка ИИ

Для настроенного проверочного экземпляра выполняются живые сценарии RouterAI:

```
cd /opt/obuchai-rf/deploy
docker compose exec -T backend python scripts/smoke_routerai.py
docker compose exec -T backend python scripts/smoke_routerai.py --images
```

Проверяются быстрые и сложные текстовые задачи, независимая проверка фактов, OCR/vision, веб-поиск, генерация и доработка изображений. HTTP healthcheck сам по себе доступность ИИ не подтверждает.

7. ОБНОВЛЕНИЕ И ЖУРНАЛЫ

Перед обновлением создаётся и проверяется резервная копия. Новые исходники и образы должны относиться к одной версии поставки. Готовую БД повторно не инициализируют; новые миграции применяются отдельно. После обновления повторяются smoke.sh и remote_check.py.

Журналы контейнеров просматриваются через docker compose logs. JSON-журналы ограничены: 100 МБ × 10 файлов на контейнер. Журналы Caddy находятся в deploy/logs/caddy. Перед публикацией журналов их необходимо проверить на наличие идентификаторов и иных непубличных данных.

8. РЕЗЕРВНОЕ КОПИРОВАНИЕ

Создание копии: sudo /opt/obuchai-rf/scripts/backup.sh. Каталог /var/backups/obuchai-rf/<UTC-дата> содержит postgres.dump, minio.tar.gz, runtime.env и SHA256SUMS.

Проверка: sudo /opt/obuchai-rf/scripts/verify_backup.sh /var/backups/obuchai-rf/<UTC-дата>. Сценарий сверяет SHA-256, восстанавливает дамп во временную БД, проверяет три аккаунта, темы, демонстрацию и файлы MinIO, после чего удаляет временную БД.

Копия содержит секреты и не подлежит публикации. Остановка без удаления данных: cd /opt/obuchai-rf/deploy && docker compose stop. Удаление volumes допускается только после подтверждённой копии и решения администратора.

9. ДОСТУП И ЛИЦЕНЗИОННЫЕ МЕХАНИЗМЫ

Программа предоставляется как веб-сервис и не использует внешние серверы активации или лицензионные ключи. Доступ пользователя определяется локальной учётной записью, JWT-сессией и тарифом в PostgreSQL проверочного экземпляра. Администрирование выполняется по SSH отдельным ключом эксперта.

Средства управления доступом находятся на том же сервере в РФ и контролируются правообладателем. Платёжная проверка тарифа в проверочном контуре заменена локальной выдачей тестового доступа без денежных операций.

10. ТИПОВЫЕ НЕИСПРАВНОСТИ

– Не открывается приложение — проверить DNS и HTTPS, реквизиты веб-шлюза и аккаунта, docker compose ps, оба Caddy и сеть public_app; для альтернативного доступа проверить SSH-туннель и привязку 127.0.0.1:18080.

– Не открывается лендинг или приложение по HTTPS — проверить A-запись домена, порты 80/443, контейнер obuchai-rf-public-site, внутренний Caddy и журнал выпуска TLS-сертификата.

– API возвращает 502 — проверить backend и его журналы.

– Ошибка PostgreSQL — проверить healthcheck, DB_* и свободное место; не перезаписывать существующую БД первичной инициализацией.

– Не формируется PPTX — проверить presentations, INTERNAL_API_KEY и PRESENTATIONS_SERVICE_URL.

– Не работают ИИ-функции — проверить действующий RouterAI-ключ, баланс и результат smoke_routerai.py.

– Не загружаются файлы — проверить MinIO, MINIO_* и свободное место.

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного документа и дата	Подп.	Дата
Изм.	измененных	заменен-ных	новых	аннулированных					