

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ (РАЗВЕРТЫВАНИЮ)
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**«Информационная система обеспечения образовательной деятельности
“Гармония Образования”»**

Содержание

1. Введение	3
2. Назначение программного обеспечения	3
3. Состав программного обеспечения.....	4
4. Порядок предоставления доступа к экземпляру программного обеспечения.....	5
5. Требования к среде развертывания.....	6
5.1. Аппаратные требования.....	6
5.2. Программные требования.....	6
5.3. Сетевые требования	6
6. Состав поставки и структура каталогов	7
6.1. Основные каталоги модулей	7
6.2. Типовой состав каталога отдельного модуля	7
7. Общий порядок установки модулей.....	8
7.1. Подготовка сервера	8
7.2. Размещение файлов модуля.....	8
7.3. Установка зависимостей	8
7.4. Подготовка переменных окружения.....	8
7.5. Импорт базы данных	9
8. Настройка базы данных	9
9. Настройка конфигурации окружения (.env)	9
10. Настройка веб-сервера Nginx	10
10.1. Типовой шаблон Nginx для серверных модулей.....	10
10.2. Размещение конфигурации Nginx.....	11
11. Запуск модулей и проверка работоспособности	11
11.1. Запуск модулей	11
11.2. Контроль состояния.....	11
11.3. Проверка работоспособности.....	12
12. Особенности развертывания модуля «Расписание»	12
12.1. Конфигурация Nginx для модуля «Расписание»	12
12.2. Порядок развертывания модуля «Расписание»	13
13. Журналирование и диагностика.....	13
14. Обновление программного обеспечения.....	14
15. Удаление программного обеспечения.....	14
16. Контактная информация	15
Примечание	16

1. Введение

Настоящий документ устанавливает порядок установки, настройки, запуска и проверки работоспособности программного обеспечения «Информационная система обеспечения образовательной деятельности «Гармония Образования»» (далее — Программное обеспечение, ПО).

Документ подготовлен для предоставления экспертам в рамках процедуры рассмотрения заявления о включении сведений о программном обеспечении в единый реестр российских программ для электронных вычислительных машин и баз данных.

ПО представляет собой комплекс взаимосвязанных веб-модулей, разворачиваемых на серверной инфраструктуре под управлением операционной системы семейства Linux и предоставляемых пользователям в формате интернет-сервиса.

Правообладатель готов предоставить экспертам доступ к экземпляру Программного обеспечения, исходным компонентам, конфигурации окружения и сопутствующим материалам, необходимым для проверки установки и работоспособности системы.

2. Назначение программного обеспечения

Программное обеспечение «Информационная система обеспечения образовательной деятельности «Гармония Образования»» предназначено для автоматизации процессов сопровождения образовательной деятельности, централизованного предоставления цифровых сервисов участникам образовательного процесса и обеспечения доступа к учебной, организационной и справочной информации.

ПО обеспечивает пользователям следующий функционал:

- централизованный доступ к сервисам системы через единый образовательный портал;
- предоставление информации и выполнение пользовательских сценариев через голосовой интерфейс;
- проведение анонимных опросов и голосований;
- бронирование, учет выдачи и возврата оборудования;
- публикация новостей, уведомлений и событий;

- формирование, редактирование и публикация актуального учебного расписания;
- формирование отчетов и аналитических сведений о ходе образовательного процесса.

Функционал различается в зависимости от роли пользователя. В ПО реализованы следующие роли:

- обучающиеся;
- родители (законные представители);
- педагогические работники;
- административные сотрудники;
- технические администраторы системы.

3. Состав программного обеспечения

В состав Программного обеспечения входят следующие основные модули:

1. **Модуль «Единый образовательный портал»** — единая точка входа к сервисам системы и пользовательским разделам.
2. **Модуль «Голосовой помощник»** — серверная часть интеграции с голосовым интерфейсом.
3. **Модуль «Анонимное голосование»** — создание, публикация и обработка анонимных опросов.
4. **Модуль «Аренда ноутбуков»** — учет и бронирование оборудования.
5. **Модуль «Телеграм-бот и Лента событий»** — авторизация и получение прав доступа, просмотр расписания, публикация новостей, уведомлений и событий.
6. **Модуль «Расписание»** — предоставление и обновление учебного расписания.
7. **Модуль «Прогресс-репорт»** — формирование отчетов и аналитических материалов.
8. **Служебные компоненты** — база данных, конфигурационные файлы, SQL-скрипты и настройки веб-сервера.

Архитектурно программное обеспечение реализовано как набор серверных приложений на платформе Node.js. Для публикации HTTP(S)-доступа используется веб-сервер Nginx. Для хранения данных используется реляционная

база данных. Развертывание отдельных модулей выполняется по единообразной схеме.

4. Порядок предоставления доступа к экземпляру программного обеспечения

Правообладатель готов предоставить экспертам доступ к экземпляру Программного обеспечения одним из следующих способов:

- удаленный доступ к развернутому тестовому экземпляру на сервере правообладателя;
- предоставление архива с исходными файлами и инструкцией по развертыванию;
- предоставление конфигурационных файлов, SQL-скриптов, параметров запуска и шаблонов веб-сервера для самостоятельного развертывания;
- демонстрация работоспособности модулей в тестовой среде правообладателя.

Учетные данные для подключения, адрес сервера, список модулей и иные сведения, необходимые для проведения проверки, передаются правообладателем уполномоченному эксперту.

Эксперт может получить доступ к экземпляру программного обеспечения, выполнив следующие действия:

1. Доступ с помощью SSH, команда вида:
 - a. Host: platon.teyhd.ru
 - b. Port: 9022
 - c. Login: potester
 - d. Pass: platon2023

2. После успешной авторизации на сервере открывается доступ к модулям экземпляра программного обеспечения. Компоненты расположены на этом же сервере. Ниже в описании компонентов для каждого компонента приводится его назначение.

Контактные данные для согласования доступа и демонстрации:

- телефон: +7 960 371-92-71;
- электронная почта: vladteyhd@mail.ru.

5. Требования к среде развертывания

5.1. Аппаратные требования

Минимальные требования для развертывания одного или нескольких модулей:

- процессор: от 2 вычислительных ядер;
- оперативная память: от 4 ГБ;
- свободное дисковое пространство: от 20 ГБ;
- сетевой доступ к HTTP/HTTPS-портам и к порту базы данных при раздельном размещении.

Рекомендуемые требования:

- процессор: от 4 вычислительных ядер;
- оперативная память: от 8 ГБ;
- свободное дисковое пространство: от 50 ГБ;
- выделенный сервер или виртуальная машина с постоянным сетевым подключением.

5.2. Программные требования

Для установки и запуска Программного обеспечения требуется:

- операционная система Linux серверного класса;
- Node.js;
- npm;
- PM2 либо иной совместимый менеджер процессов Node.js;
- веб-сервер Nginx;
- СУБД MySQL или совместимая реляционная СУБД;
- доступ к SSL-сертификатам при публикации по HTTPS.

5.3. Сетевые требования

Для публикации модулей в сети могут использоваться доменные имена второго и третьего уровня. Каждый модуль может публиковаться:

- на отдельном поддомене;
- на отдельном серверном порту за обратным прокси Nginx;
- как статическое приложение и серверный API одновременно.

6. Состав поставки и структура каталогов

6.1. Основные каталоги модулей

Экземпляр ИСООД «Гармония Образования» размещается в файловой системе сервера. Типовая директория размещения модулей:

```
/var/www/html/
```

В указанной директории располагаются каталоги модулей, в том числе:

```
/portal_platoniks - модуль «Единый образовательный портал»  
/alisa_skill_platoniks - модуль «Голосовой помощник»  
/laptop_management - модуль «Аренда ноутбуков»  
/telegram_bot_and_event_feed - модуль «Лента событий»  
/school_schedule - модуль «Расписания»  
/anonym_voting – модуль «Анонимного голосования»  
/progress_report – модуль «Прогресс-репортов»
```

При наличии отдельных каталогов других модулей следует размещать их по аналогичной схеме в указанной корневой директории.

6.2. Типовой состав каталога отдельного модуля

Стандартный набор файлов и подкаталогов, входящих в состав модуля:

```
<module>/  
├── package.json - описание зависимостей и команд сборки/запуска  
├── package-lock.json - зафиксированные версии зависимостей  
├── .env - параметры окружения модуля  
├── SQL/ - SQL-скрипты для создания и инициализации БД  
├── nginx/ - шаблоны конфигурации Nginx (при наличии)  
├── public/ - статические ресурсы  
├── dist/ - результирующие файлы клиентской сборки (при наличии)  
├── src/ - исходные тексты  
├── logs/ - журналы работы (при наличии)  
└── ecosystem.config.js - конфигурация PM2 (при наличии)
```

Фактический состав файлов конкретного модуля может отличаться, однако общий порядок развертывания остается единым: установка зависимостей, заполнение файла .env, импорт базы данных из каталога SQL, запуск модуля и публикация его через Nginx.

7. Общий порядок установки модулей

7.1. Подготовка сервера

Перед установкой необходимо:

1. Подготовить сервер под управлением Linux.
2. Установить Node.js и npm.
3. Установить Nginx.
4. Установить СУБД MySQL или совместимую СУБД.
5. Установить PM2.
6. Создать рабочие каталоги в /var/www/html/.
7. Обеспечить права доступа пользователя, от имени которого выполняется запуск приложений.

7.2. Размещение файлов модуля

Файлы соответствующего модуля размещаются в целевой директории, например:

```
/portal_platoniks  
/alisa_skill_platoniks  
/laptop_management  
/telegram_bot_and_event_feed  
/school_schedule  
/anonym_voting  
/progress_report
```

7.3. Установка зависимостей

В каталоге каждого модуля выполняется установка зависимостей Node.js:

```
cd /var/www/html/<module>  
npm install
```

При использовании зафиксированных версий зависимостей допускается применение команды:

```
npm ci
```

7.4. Подготовка переменных окружения

После размещения файлов необходимо создать и заполнить файл .env для каждого модуля. Файл .env должен содержать параметры подключения к базе данных, сетевые порты, адреса внешних сервисов, секреты приложений и иные

параметры, требуемые конкретным модулем. Пример лежит в папке с проектом `.env.example`.

7.5. Импорт базы данных

После настройки `.env` необходимо инициализировать базу данных SQL-скриптами из SQL соответствующего модуля. Общий порядок импорта:

```
mysql -u <db_user> -p <db_name> < /var/www/html/ SQL/<file>.sql
```

При наличии нескольких SQL-файлов они выполняются в порядке, предусмотренном структурой поставки соответствующего модуля.

8. Настройка базы данных

Для каждого модуля, использующего отдельную базу данных либо отдельную логическую схему, необходимо:

1. Создать базу данных.
2. Создать учетную запись пользователя базы данных.
3. Назначить права доступа этой учетной записи.
4. Импортировать SQL-скрипты из каталога SQL.
5. Указать параметры подключения в `.env`.

Типовой перечень параметров, используемых в `.env`:

- `DB_HOST` — адрес сервера базы данных;
- `DB_PORT` — порт подключения;
- `DB_NAME` — имя базы данных;
- `DB_USER` — пользователь базы данных;
- `DB_PASSWORD` — пароль пользователя базы данных.

При необходимости могут использоваться дополнительные параметры, предусмотренные логикой конкретного модуля.

9. Настройка конфигурации окружения (`.env`)

Перед запуском каждого модуля необходимо заполнить файл `.env`. Типовая конфигурация может включать:

```
PORT=777
DB_HOST=127.0.0.1
DB_PORT=3306
DB_NAME=app_db
```

```
DB_USER=app_user
DB_PASSWORD=secret
NODE_ENV=production
```

Дополнительно в зависимости от модуля могут применяться:

- параметры авторизации;
- ключи интеграции с внешними сервисами;
- адреса API;
- доменные имена;
- параметры журналирования;
- настройки токенов и сессий.

Фактический перечень переменных окружения определяется содержимым конкретного модуля.

10. Настройка веб-сервера Nginx

Для публикации модулей используется Nginx. Для большинства серверных модулей применяется единый шаблон обратного прокси.

10.1. Типовой шаблон Nginx для серверных модулей

```
server {
server_name platoniks.ru;

location / {
proxy_pass http://localhost:777/;
proxy_set_header Host $host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header nip $remote_addr;
}

listen 443 ssl;
ssl_certificate /etc/letsencrypt/live/platoniks.ru/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/platoniks.ru/privkey.pem;
include /etc/letsencrypt/options-ssl-nginx.conf;
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}
```

При использовании данного шаблона необходимо:

- заменить `server_name` на доменное имя соответствующего модуля;

- заменить порт 777 на фактический порт модуля;
- убедиться в наличии корректных SSL-сертификатов.

10.2. Размещение конфигурации Nginx

Конфигурационный файл модуля размещается в каталоге конфигурации Nginx, например:

```
/etc/nginx/sites-available/<module>.conf
```

Затем создается символическая ссылка:

```
/etc/nginx/sites-enabled/<module>.conf
```

После добавления конфигурации выполняется проверка и перезапуск Nginx:

```
nginx -t
systemctl reload nginx
```

11. Запуск модулей и проверка работоспособности

11.1. Запуск модулей

Запуск модулей выполняется как обычных Node.js-приложений. Для долговременной работы рекомендуется использовать PM2.

Пример запуска:

```
cd /var/www/html/<module>
pm2 start npm --name <module> -- start
```

Либо, при наличии файла конфигурации PM2:

```
pm2 start ecosystem.config.js
```

11.2. Контроль состояния

Проверка состояния выполняется командой:

```
pm2 status
```

Для отдельных модулей могут использоваться команды вида:

```
pm2 status portal_platoniks
pm2 status alisa_skill_platoniks
pm2 status laptop_management
pm2 status telegram_bot_and_event_feed
pm2 status school_schedule
```

```
pm2 status anonym_voting
pm2 status progress_report
```

11.3. Проверка работоспособности

После запуска необходимо убедиться, что:

- процесс модуля находится в состоянии online в PM2;
- приложение отвечает на локальном порту;
- Nginx корректно проксирует запросы на приложение;
- веб-интерфейс модуля открывается по соответствующему доменному имени;
- модуль имеет доступ к базе данных;
- основные сценарии использования выполняются без ошибок.

12. Особенности развертывания модуля «Расписание»

Модуль «Расписание» публикуется по отдельной схеме.

12.1. Конфигурация Nginx для модуля «Расписание»

```
server {
server_name rasp.platoniks.ru;

root /var/www/html/school_schedule/dist;
index index.html;

location / {
    try_files $uri $uri/ /index.html;
}

location /api/ {
    proxy_pass http://localhost:3001;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}

listen 443 ssl;
ssl_certificate /etc/letsencrypt/live/rasp.platoniks.ru/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/rasp.platoniks.ru/privkey.pem;
include /etc/letsencrypt/options-ssl-nginx.conf;
```

```
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;  
}
```

12.2. Порядок развертывания модуля «Расписание»

1. Разместить клиентскую часть модуля в каталоге `/var/www/html/school_schedule`.
 2. Разместить серверную часть модуля в каталоге `/var/www/html/school_schedule/backend`.
 3. Выполнить установку зависимостей в каталоге серверной и клиентской частей.
 4. Заполнить `.env` для серверной части.
 5. Импортировать SQL-скрипты из каталога `SQL`, относящегося к серверной части.
6. Выполнить сборку клиентской части так, чтобы результирующие файлы были размещены в каталоге `dist`.

Для запуска модуля расписания:

1. `cd /var/www/html/school_schedule`
 2. `npm run build`
 3. `cd /backend`
 4. `pm2 start npm --name school_schedule -- start`
7. Запустить серверную часть на локальном порту 3001 либо на ином порту, указанном в конфигурации.
8. Подключить и активировать конфигурацию Nginx.
9. Выполнить проверку открытия веб-интерфейса и ответа API.

13. Журналирование и диагностика

Для диагностики и контроля работы системы используются:

- Журналы PM2;
- Журналы Nginx;
- Журналы операционной системы;
- Прикладные логи, формируемые модулями.

Типовые команды диагностики:

```
pm2 logs  
journalctl -u nginx  
cat /var/log/nginx/error.log  
cat /var/log/nginx/access.log
```

При наличии прикладного каталога логов в конкретном модуле могут использоваться файлы из каталога logs/.

В случае ошибок необходимо последовательно проверить:

1. корректность заполнения .env;
2. доступность базы данных;
3. корректность импорта SQL-скриптов;
4. доступность локального порта приложения;
5. корректность конфигурации Nginx;
6. наличие прав доступа к рабочим каталогам и файлам.

14. Обновление программного обеспечения

Обновление отдельных модулей выполняется по следующему порядку:

1. Создать резервную копию базы данных и конфигурационных файлов.
2. Остановить модуль либо перевести его в режим обслуживания.
3. Обновить файлы модуля в рабочем каталоге.
4. При необходимости обновить зависимости командой `npm install` или `npm ci`.
5. При наличии новых SQL-скриптов выполнить обновление структуры базы данных.
6. Проверить файл .env на актуальность параметров.
7. Перезапустить модуль через PM2.
8. Выполнить проверку работоспособности.

Пример перезапуска:

```
pm2 restart <module>
```

15. Удаление программного обеспечения

Удаление модуля выполняется в следующем порядке:

1. Остановить процесс модуля.
2. Удалить конфигурацию Nginx для модуля.
3. Удалить каталог модуля из файловой системы.
4. При необходимости удалить базу данных модуля.
5. Перезагрузить или перечитать конфигурацию Nginx.

Пример:

```
pm2 delete <module>
rm -rf /var/www/html/<module>
systemctl reload nginx
```

Удаление выполняется администратором системы с учетом требований по сохранности данных и резервному копированию.

16. Контактная информация

Правообладатель готов предоставить дополнительную информацию, консультации по развертыванию и доступ к тестовому экземпляру программного обеспечения.

Контактные данные:

- телефон: +7 960 371-92-71;
- электронная почта: vladteyhd@mail.ru.

Фактический адрес разработки, размещения разработчиков и службы поддержки пользователей:

197110, г. Санкт-Петербург, Футбольная аллея, д. 8, литера А, помещ. 1Н (часть), помещ. 114–143.

Лицензии библиотек, фреймворков, языков программирования

Лицензии библиотек, фреймворков, языков программирования, использованных при разработке компонентов сервиса “Гармония образования”, разработанных ООО «Платоникс»

Наименование компонента	Наименование лицензии	Ссылка на лицензию	Ссылка на репозиторий
Nginx 1.13.6.1	2-пунктная лицензия BSD	https://nginx.org/LICENSE	https://github.com/nginx/nginx

Node.js	MIT License	https://github.com/nodejs/node/blob/main/LICENSE	https://github.com/nodejs/node
MySQL	GPL v2	https://gnu.org/licenses/old-licenses/gpl-2.0.html	https://github.com/mysql/mysql-server
Vite	MIT License	https://github.com/vitejs/vite/blob/main/LICENSE	https://github.com/vitejs/vite
React	MIT License	https://github.com/facebook/react/blob/main/LICENSE	https://github.com/facebook/react
plugNmeet	MIT License	https://github.com/mynaparrot/plugNmeet-server/blob/main/LICENSE	https://github.com/mynaparrot/plugNmeet-server
KodExplorer	GNU GPL v3.0	https://kodcloud.com/license	https://github.com/kalca ddle/KodExplorer

Примечание

В случае предоставления экспертам удаленного доступа к экземпляру ПО правообладатель дополнительно передает:

- адрес сервера;
- учетные данные для подключения;
- перечень доступных модулей;
- тестовые учетные записи пользователей;
- сведения о маршрутах доступа к веб-интерфейсам и API.