Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДАЮ | |  | УТВЕРЖДАЮ | |
| Д.Г. Демидов (декан, Московский политехнический университет) | |  | Е.В. Шукалова (старший преподаватель кафедры Инфокогнитивных технологий, Московский политехнический университет) | |
| Личная  подпись | Расшифровка подписи |  | Личная  подпись | Расшифровка подписи |
| Печать | |  | Печать | |
| Дата | |  | Дата | |

Автоматизированная информационная система

Система профессиональной подготовки школьников

АИС «Пора»

**ОПИСАНИЕ СИСТЕМЫ**

На 20 листах

Действует с 30.12.2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СОГЛАСОВАНО | |  |  | |
| М.В. Даньшина (заместитель декана, Московский политехнический университет) | |  |  | |
| Личная  подпись | Расшифровка подписи |  |  |  |
| Печать | |  |  | |
| Дата | |  |  | |

**СОДЕРЖАНИЕ**

[1 ОБЩИЕ СВЕДЕНИЯ 3](#_Toc91712267)

[1.1 Полное наименование системы и ее условное обозначение 3](#_Toc91712268)

[1.2 Требования к программному обеспечению 3](#_Toc91712269)

[1.3 Языки программирования 3](#_Toc91712270)

[2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ 4](#_Toc91712271)

[2.1 Назначение системы 4](#_Toc91712272)

[2.2 Цели создания системы 4](#_Toc91712273)

[3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ 7](#_Toc91712274)

[3.1 Алгоритм программы 7](#_Toc91712275)

[3.2 Методы решения 10](#_Toc91712276)

[3.3 Структура программы 10](#_Toc91712277)

[4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА 12](#_Toc91712278)

[5 ВЫЗОВ И ЗАГРУЗКА 14](#_Toc91712279)

[5.1 Запуск в полноценной контейнеризованной среде 14](#_Toc91712280)

[5.2 Запуск в контейнеризованной среде разработки 14](#_Toc91712281)

[5.3 Запуск модулей программы в неконтейнеризованном окружении 15](#_Toc91712282)

[6 ВХОДНЫЕ ДАННЫЕ 17](#_Toc91712283)

[7 ВЫХОДНЫЕ ДАННЫЕ 19](#_Toc91712284)

# ОБЩИЕ СВЕДЕНИЯ

## Полное наименование системы и ее условное обозначение

Полное наименование системы: Система профессиональной подготовки школьников «Пора».

Краткое наименование системы: АИС «Пора».

## Требования к программному обеспечению

Для взаимодействия с АИС «Пора» необходимо использовать веб-браузер. Рекомендуемые веб-браузеры – Google Chrome версии 96.0 и выше или Mozilla Firefox 95.0 и выше. Рекомендуемые операционные системы Windows 10 или macOS Sierra, для мобильных устройств – Android 8.1 или   
iOS 11.

## Языки программирования

АИС «Пора» реализована на основе фронтенд-фреймворка Vue.js, языка программирования Python, фреймворка Django и системы управления базами данных PostgreSQL. Адресация и доступ к модулям регулируются веб-сервером Nginx.

Для построения клиент-серверного взаимодействия использована технология Apollo GraphQL.

Для контейнеризации микросервисов, их запуска и масштабирования использованы технологии Docker и Docker Compose.

# ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

## Назначение системы

АИС «Пора» предназначена для информирования школьников о проведении образовательных курсов высшими учебными заведениями, а также для проведения компаниями олимпиадных мероприятий для школьников в формате онлайн, которое включает в себя следующие процессы:

* внесение и хранение данных об олимпиадных заданиях представителями компаний;
* внесение и хранение данных об ответах школьников на олимпиадные задания;
* внесение и хранение данных о проверке представителями компаний ответов школьников;
* автоматическое создание сертификата об успешном прохождении олимпиадного задания школьником.

## Цели создания системы

Основными целями создания АИС «Пора» являются:

* повышение эффективности поиска школьниками информации о проведении образовательных курсов высшими учебными заведениями России через агрегацию этих данных;
* повышение лояльности школьников к потенциальным будущим компаниям-работодателям через выполнение олимпиадных заданий от последних;
* повышение лояльности школьников к высшим учебным заведениям через предоставление информации о проведении ими образовательных курсов.

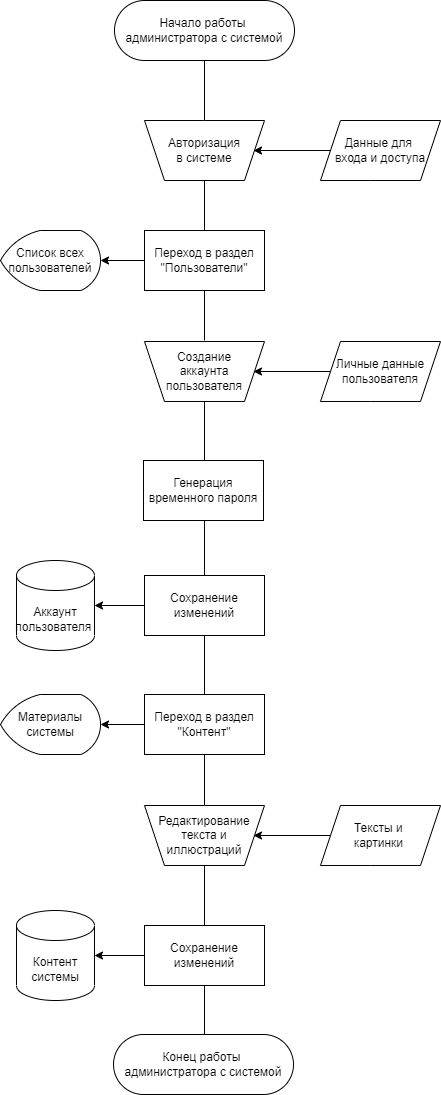
Для реализации поставленных целей система должна решать следующие задачи:

* регистрация и авторизация пользователей различных ролей с разным уровнем доступа (школьник, представитель высших учебных заведений, представитель компаний-работодателей, администратор);
* проведение опроса школьника об интересующих его предметах и направлениях;
* формирование рекомендаций курсов и олимпиадных заданий на основе результатов опроса школьника о предметах и направлениях;
* заполнение и последующее изменение данных профиля пользователей различных ролей (школьник, представитель высших учебных заведений, представитель компаний-работодателей);
* ввод данных о проведении образовательных курсов представителями высших учебных заведений;
* просмотр данных о проведении образовательных курсов представителями высших учебных заведений школьником с возможностью поиска и фильтрации;
* запись школьника на образовательный курс, проводимый высшими учебными заведениями;
* просмотр данных о записавшихся на образовательный курс школьниках представителями высших учебных заведений;
* ввод данных об олимпиадных заданиях представителями компаний-работодателей;
* ввод данных решений олимпиадных заданий школьником;
* ввод данных об оценивании решений олимпиадных заданий представителями компаний-работодателей;
* создание сертификата об успешном прохождении школьником олимпиадного задания;
* выгрузка сертификата об успешном прохождении школьником олимпиадного задания;
* просмотр данных о школьниках, решавших олимпиадное задание, представителями компаний-работодателей с возможностью поиска и фильтрации;
* просмотр данных обо всех курсах школьника, а также обо всех олимпиадных заданиях, решенных школьником, с возможностью поиска и фильтрации;
* создание новых пользователей различных ролей (школьник, представитель высшего учебного заведения, представитель компании-работодателя, администратор) администратором;
* просмотр списка всех пользователей администратором с возможностью поиска и фильтрации;
* редактирование контента главной страницы администратором.

# ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

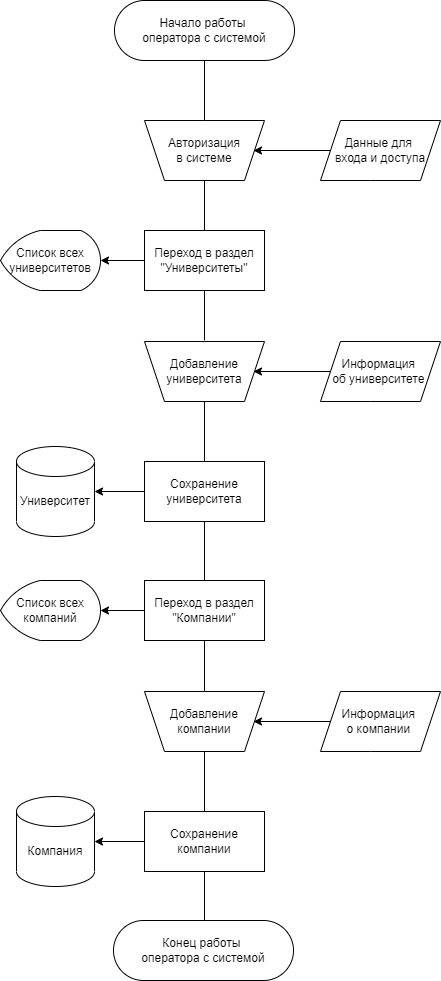
## Алгоритм программы

Администратор переходит в раздел администрирования системы, авторизуется в системе, заполняя данные почты и пароля. Администратор переходит в раздел «Пользователи». Далее он создает новый аккаунт для пользователя, выбирает его роль и назначает ему временный пароль, после чего сохраняет все изменения. Администратору доступен для просмотра и редактирования список всех зарегистрированных пользователей. Администратор переходит в раздел «Контент», добавляет иллюстрации и редактирует текст для дальнейшего размещения на главной странице системы. Алгоритм взаимодействия администратора с системой представлен на [Рисунке 1](#Рис1).



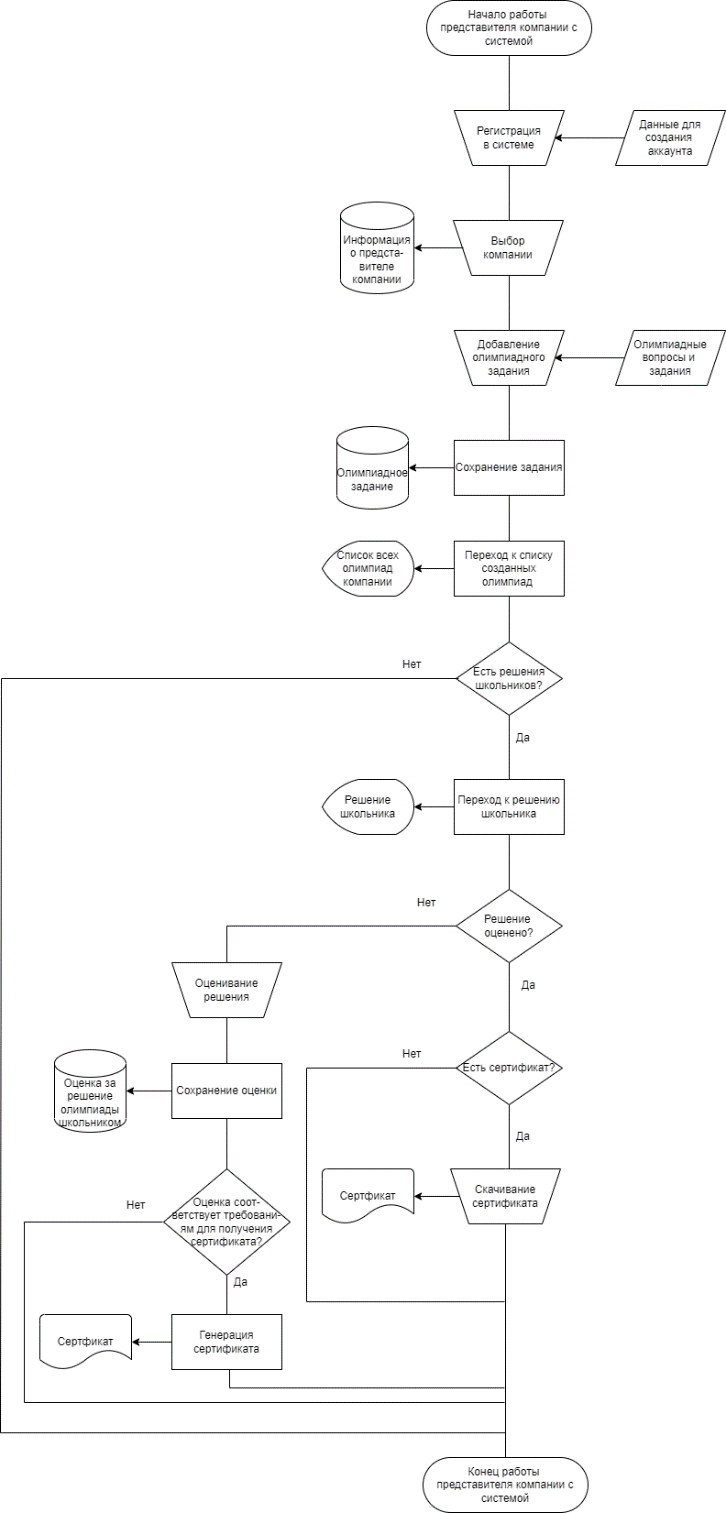
[Рисунок 1 – Модель взаимодействия администратора с системой](#Рис1текст)

Оператор переходит в раздел администрирования системы, авторизуется в системе, заполняя данные почты и пароля. Оператор переходит в раздел «Университеты». Далее он создает университет и добавляет информацию о нем, после чего сохраняет все изменения. Оператор переходит в раздел «Компании», добавляет новую компанию и информацию о ней, после чего сохраняет изменения. Алгоритм взаимодействия оператора с системой представлен на [Рисунке 2](#Рис2).



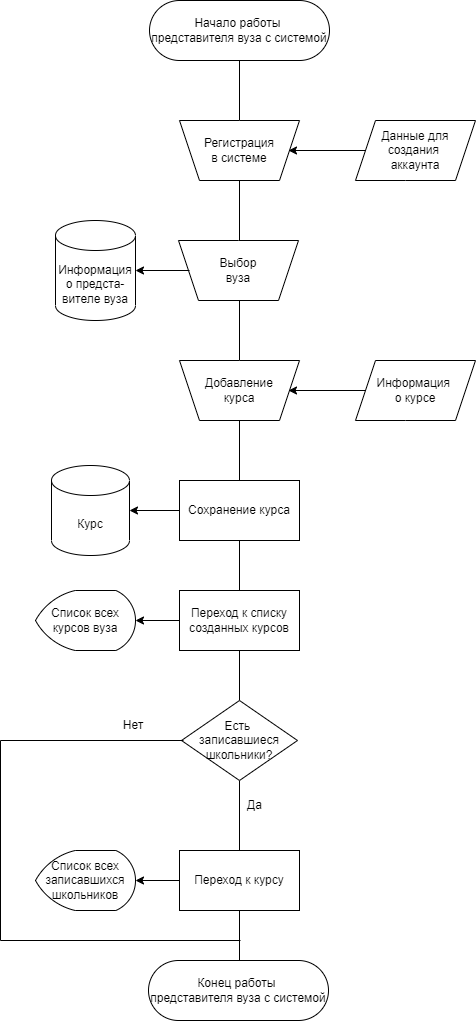
[Рисунок 2 – Модель взаимодействия оператора с системой](#Рис2текст)

Представитель компании-работодателя регистрируется в системе, заполняя данные своего профиля и прикрепляя его к одной из имеющихся в системе компаний. Далее он добавляет олимпиадное задание, настраивая его тему, школьный предмет, ограничение по времени и другие параметры, и сохраняет его. После сохранения задания представителю компании-работодателя становятся доступны к просмотру список студентов, прошедших задание, перечень их ответов, и появляется возможность проверять ответы школьников и выставлять оценку. Алгоритм взаимодействия представителя компании-работодателя с системой представлен на [Рисунке 3](#Рис3).



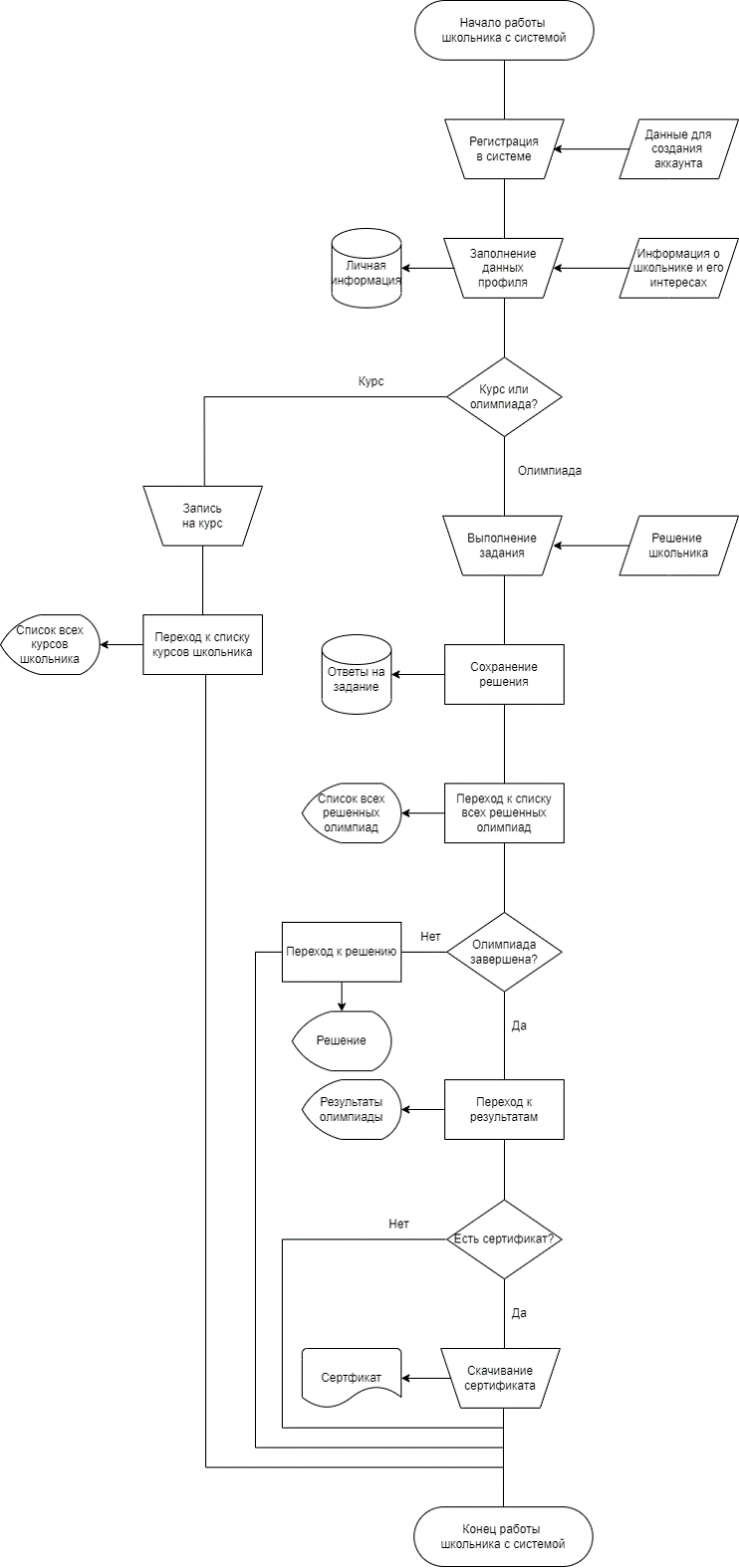
[Рисунок 3 – Модель взаимодействия представителя компании-работодателя с системой](#Рис3текст)

Представитель высшего учебного заведения регистрируется в системе, заполняя данные своего профиля и прикрепляя его к одному из имеющихся в системе высших учебных заведений. Далее он добавляет курс, настраивая тему обучения, длительность, формат проведения курса и другие параметры, и сохраняет его. После сохранения курса представителю высшего учебного заведения становится доступным к просмотру список студентов, записавшихся на курс. Алгоритм взаимодействия представителя высшего учебного заведения с системой представлен на [Рисунке 4](#Рис4).



[Рисунок 4 – Модель взаимодействия представителя высшего учебного заведения с системой](#Рис4текст)

Школьник регистрируется в системе, заполняя данные своего профиля и проходя опрос об интересующих его предметах. После прохождения опроса ему доступны рекомендации курсов и олимпиадных заданий в соответствии с выбранными им дисциплинами. Школьник может перейти на вкладку «Курсы» и выбрать курс на основе рекомендаций системы или самостоятельно, пользуясь фильтрацией и поиском. Выбрав курс, он может записаться на него, после чего курс появится в разделе «Мои курсы» в личном кабинете школьника. Школьник может перейти на вкладку «Олимпиадные задания» и выбрать задание на основе рекомендаций системы или самостоятельно, пользуясь фильтрацией и поиском. Выбрав задание, он может выполнить его, после чего задание появится в разделе «Мои задания» в личном кабинете школьника. После того, как задание будет проверено представителем компании-работодателя, в разделе «Мои задания» оно приобретет статус «Проверено». При успешном прохождении задания школьник сможет посмотреть и скачать соответствующий сертификат. Алгоритм взаимодействия школьника с системой представлен на [Рисунке 5](#Рис5).



[Рисунок 5 – Модель взаимодействия школьника с системой](#Рис5текст)

## Методы решения

В основе реализации АИС «Пора» лежит клиент-серверное взаимодействие с использованием микросервисной архитектуры, состоящей из подсистем-модулей и системы управления базами данных. Для запуска и масштабирования микросервисов внедрена контейнеризация. Адресация и доступ к модулям регулируются веб-сервером.

Интерфейсная часть системы разработана с использованием свободно распространяемого фронтенд-фреймворка.

## Структура программы

Архитектура АИС «Пора» является модульной и микросервисной. Модули представлены следующими подсистемами:

* подсистема авторизации выполняет действия по авторизации, регистрации и аутентификации. Для авторизации подсистема использует данные об учётных записях, к которым отправляет запрос в зашифрованном виде со стороны клиентской части. На сервере происходит обработка запроса и возврат данных или ответов в виде статусов;
* подсистема хранения данных выполняет хранение оперативных данных системы о сессиях и пользователе. Сохранность и целостность данных на сервере должна быть обеспечена периодическим резервным копированием, а также периодическим копированием указанных выше данных из оперативной памяти в энергонезависимую;
* подсистема олимпиадного тестирования выполняет действия по созданию, удалению, редактированию и хранению данных об олимпиадах, их заданиях и школьниках, выполняющих их, а также обеспечивает ограниченную по времени загрузку решений заданий и создание сертификата по шаблону с помощью получения данных от сервера. На клиенте формируется запрос с использованием интерфейса АИС. На сервере происходит обработка запроса и возврат данных или ответов в виде статусов;
* подсистема образовательных курсов выполняет действия по созданию, удалению, редактированию и хранению данных о курсах и школьниках, записавшихся на них. На клиенте формируется запрос с использованием интерфейса АИС. На сервере происходит обработка запроса и возврат данных или ответов в виде статусов;
* подсистема управления пользователями выполняет действия по созданию, удалению, редактированию и хранению данных аккаунтов пользователей. Подсистема обеспечивает взаимодействие с данными аккаунтов в зашифрованном виде со стороны клиентской части. На сервере происходит обработка запроса и возврат данных или ответов в виде статусов.

# ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

В составе комплекса технического обеспечения должны присутствовать следующие технические средства:

* серверы БД;
* веб-сервер;
* ПК пользователей;
* ПК администраторов;
* мобильные устройства пользователей;
* мобильные устройства администраторов.

Серверы БД должны быть объединены в отказоустойчивый кластер. Серверы БД и веб-сервер должны быть объединены одной локальной сетью с пропускной способностью не менее 30 Мбит.

Требования к техническим характеристикам серверов БД:

* процессор – 2 х Intel Xeon 3 ГГц;
* объем оперативной памяти – 16 Гб;
* дисковая подсистема – 4 х 146 Гб;
* сетевой адаптер – 30 Мбит/с.

Требования к техническим характеристикам веб-сервера:

* процессор – 2 х Intel Xeon 3 ГГц;
* объем оперативной памяти – 16 Гб;
* дисковая подсистема – 4 х 146 Гб;
* сетевой адаптер – 30 Мбит/с.

Требования к техническим характеристикам ПК пользователя и администратора:

* процессор – Intel Pentium 1.5 ГГц;
* объем оперативной памяти – 256 Мб;
* видеокарта – 256 Мб;
* дисковая подсистема – 40 Гб;
* сетевой адаптер – 30 Мбит/с;
* монитор – разрешение не менее 300x400 пикселей;
* клавиатура;
* мышь или тачпад.

Требование к техническим характеристикам мобильных устройств – операционная система Android версии 8.1 и выше или операционная система iOS версии 11 и выше.

# ВЫЗОВ И ЗАГРУЗКА

## Запуск в полноценной контейнеризованной среде

Перед запуском программы в полноценной контейнеризованной среде необходимо установить следующие утилиты для интерфейса командной строки:

* Docker;
* Docker Compose.

Первый запуск программы в полноценной контейнеризованной среде требует стабильного интернет-подключения для установки зависимостей проекта.

Для запуска программы необходимо от имени администратора системы перейти в корневую директорию проекта и выполнить следующую команду из интерфейса командной строки: «docker-compose -f docker-compose-prod.yml up --build -d».

После завершения загрузки зависимостей и запуска контейнеров для программных модулей система станет доступна на порте, который будет указан в консоли.

Для создания пользователя для отладки с доступом к административной панели необходимо при запущенной программе перейти в корневую директорию проекта и выполнить следующую команду из интерфейса командной строки: «docker-compose run backend python manage.py createsuperuser».

## Запуск в контейнеризованной среде разработки

Перед запуском программы в контейнеризованной среде разработки необходимо установить следующие утилиты для интерфейса командной строки:

* Docker;
* Docker Compose.

Первый запуск программы в контейнеризованной среде разработки требует стабильного интернет-подключения для установки зависимостей проекта.

Для запуска программы необходимо от имени администратора системы перейти в корневую директорию проекта и выполнить следующую команду из интерфейса командной строки: «docker-compose up --build».

После завершения загрузки зависимостей и запуска контейнеров для программных модулей система станет доступна на порте, который будет указан в консоли.

Для создания пользователя для отладки с доступом к административной панели необходимо при запущенной программе перейти в корневую директорию проекта и выполнить следующую команду из интерфейса командной строки: «docker-compose run backend python manage.py createsuperuser».

## Запуск модулей программы в неконтейнеризованном окружении

### Запуск модуля визуального интерфейса системы

Перед запуском модуля серверной части необходимо установить программную платформу Node.js не ниже 16 версии.

Для запуска веб-сервера разработки интерфейса необходимо перейти из корневой директории проекта в папку frontend и выполнить следующую последовательность команд из интерфейса командной строки:

* «npm install -g vue-cli» (выполняется при первом запуске);
* «npm install»;
* «npm run serve».

После завершения загрузки зависимостей и запуска модуля интерфейс будет доступен по адресу «http://127.0.0.1:3000/», если порт 3000 не будет занят иным локальным приложением.

### Запуск модуля серверной части системы

Перед запуском модуля серверной части необходимо установить оболочку языка Python версии 3.

Для запуска модуля серверной части необходимо перейти из корневой директории проекта в папку backend и выполнить следующую команду из интерфейса командной строки: «python manage.py runserver».

После завершения загрузки зависимостей и запуска модуля серверная часть, а также интерфейс административной панели будут доступны по адресам «http://127.0.0.1:8000/» и «http://127.0.0.1:8000/admin» соответственно, если порт 8000 не будет занят иным локальным приложением.

# ВХОДНЫЕ ДАННЫЕ

Входные данные от школьника будут представлены следующим образом:

* фамилия (текстовое поле);
* имя (текстовое поле);
* отчество (текстовое поле);
* дата рождения (поле с типом «дата», представление хранится в формате ДД.ММ.ГГГГ);
* город проживания (текстовое поле);
* email адрес (текстовое поле с типом «email»);
* интересующие школьные предметы (список полей с типом «чекбокс»).

Входные данные от представителя высшего учебного заведения будут представлены следующим образом:

* фамилия (текстовое поле);
* имя (текстовое поле);
* отчество (текстовое поле);
* email адрес (текстовое поле с типом «email»);
* название высшего учебного заведения (текстовое поле);
* город высшего учебного заведения (текстовое поле);
* описание высшего учебного заведения (текстовое поле);
* название курса (текстовое поле);
* связанные с курсом школьные предметы (список полей с типом «чекбокс»);
* описание курса (текстовое поле);
* длительность курса (поле-раскрывающийся список);
* формат проведения курса (поле-раскрывающийся список);
* даты проведения курса (два поля с типом «дата», представление хранится в формате ДД.ММ.ГГГГ);
* ограничение по количеству слушателей курса (числовое поле).

Входные данные от представителя компании-работодателя будут представлены следующим образом:

* фамилия (текстовое поле);
* имя (текстовое поле);
* отчество (текстовое поле);
* email адрес (текстовое поле с типом «email»);
* название компании (текстовое поле);
* описание компании и ее деятельности (текстовое поле);
* название олимпиады (текстовое поле);
* связанные с олимпиадой школьные предметы (список полей с типом «чекбокс»);
* описание олимпиады (текстовое поле);
* время на прохождение олимпиады в минутах (числовое поле);
* задания олимпиады (текстовое поле);
* конечная дата проведения олимпиады (поле с типом «дата», представление хранится в формате ДД.ММ.ГГГГ);
* конечная дата оглашения результатов олимпиады (поле с типом «дата», представление хранится в формате ДД.ММ.ГГГГ).

Передача входных данных между подсистемами и их обработка осуществляются на основе защищенных двойным шифрованием форматов обмена данными, кодирование реализовано на основе JWT методологии.

# ВЫХОДНЫЕ ДАННЫЕ

Выходные данные для школьников будут представлены следующим образом:

* сертификаты об успешном прохождении олимпиадных заданий, доступные для скачивания в формате PDF;
* список всех пройденных курсов в формате отображаемой на экране таблицы;
* список всех решенных олимпиадных заданий в формате отображаемой на экране таблицы.

Выходные данные для представителей высших учебных заведений будут представлены в виде отображаемого на экране в виде таблицы списка всех школьников, прошедших курсы.

Выходные данные для представителей компаний-работодателей будут представлены в виде отображаемого на экране в виде таблицы списка всех школьников, прошедших олимпиадные задания, а также в виде доступных для скачивания в формате PDF сертификатов школьников об успешном прохождении олимпиадных заданий.

Передача выходных данных между подсистемами и их обработка осуществляются на основе защищенных двойным шифрованием форматов обмена данными, кодирование реализовано на основе JWT методологии.

**СОСТАВИЛИ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование организации, предприятия** | **Должность исполнителя** | **Фамилия, имя, отчество** | **Подпись** | **Дата** |
| Московский политехнический университет | Студент | Беляева Дарья Владиславовна |  | 30.12.2021 |

**СОГЛАСОВАНО**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование организации, предприятия** | **Должность исполнителя** | **Фамилия, имя, отчество** | **Подпись** | **Дата** |
| Московский политехнический университет | Заместитель декана | Даньшина Марина Владимировна |  | 30.12.2021 |