

1. Компания ООО "Интаро Софт" существует на рынке веб-разработки с 2006 года. За время существования компании сотрудники реализовали проекты для крупных государственных и коммерческих заказчиков: "Lacoste Russia", "FixPrice", "Sony Centre", Сколково, Государственная Дума и др.

"Интаро Софт" и, в частности, подразделение backend-разработки специализируются на разработке и сопровождении веб-проектов, создании веб-приложений, на внедрении CRM-систем любой сложности, их интеграции с внутренними и внешними IT-системами, а также на обеспечении технического сопровождения на всех этапах разработки ПО.

2. В рамках проекта создаётся файл `docker-compose.yml`, описывающий архитектуру из трёх контейнеров: веб-сервера Nginx, базы данных PostgreSQL и PHP-интерпретатора. Используется версия 3.1, поддерживающая актуальный синтаксис и сетевое взаимодействие. Контейнеры объединены внутренней сетью Docker, обеспечена работа с монтируемыми томами для хранения данных и конфигураций. Nginx обслуживает проект из директории `/app` и связан с PHP-контейнером, построенным из собственного Dockerfile. PostgreSQL доступна только с хоста, с безопасной настройкой и сохранением данных в именованном томе. Переменные окружения управляют портами и доступом, Composer настраивается для работы внутри PHP-контейнера. Решение обеспечивает удобную и гибкую среду для локальной разработки.
3. Когда инфраструктура уже настроена и база данных спроектирована, можно переходить к следующему важному этапу — созданию сущностей, отражающих структуру данных в приложении. Для этого в проекте используется ORM-библиотека Doctrine, которая позволяет описывать модели данных в виде классов PHP, а затем автоматически синхронизировать их с базой данных. После запуска этой команды Symfony пошагово задаёт вопросы: нужно указать название сущности, определить поля, а также задать

дополнительные параметры — например, можно ли этому полю быть пустым, или должно ли оно быть уникальным. Таким образом, происходит полуавтоматическая генерация кода, при этом разработчик сохраняет полный контроль над моделью данных.

Результатом выполнения команды становится PHP-класс сущности, находящийся в директории `src/Entity`, а также автоматически создаётся и обновляется файл миграции при последующих действиях.

4. Вопрос автоматизации процесса обучения программированию активно рассматривается в научной литературе и уже получил широкое практическое применение в виде современных образовательных платформ и систем дистанционного обучения. Среди известных решений можно выделить как специализированные платформы (Coursera, Udemy, Stepik), так и полнофункциональные системы управления обучением (LMS), например Moodle, предоставляющие инструменты для создания курсов, контроля успеваемости и автоматизированного тестирования. Такие платформы играют ключевую роль в цифровизации образовательного процесса.

5. Первая модель — **модель успеваемости студентов**.

Её цель — количественно оценить, насколько успешно студент осваивает материал курса. Для этого используется простая, но информативная метрика — **средняя оценка за выполненные задания**. То есть мы просто берём все оценки, которые получил студент, и делим их сумму на количество заданий, которые он выполнил. Таким образом, мы получаем усреднённую успеваемость по курсу.

Вторая модель — модель персонализированных рекомендаций.

Она используется для того, чтобы предлагать студенту наиболее релевантные курсы на основе его интересов и предыдущей активности. В центре модели — теги, которыми помечены курсы, и с которыми студент чаще взаимодействует. Она показывает, какие тематики студент выбирает чаще всего. Например, если студент прошёл 10 курсов, из которых 6 были помечены тегом "frontend", то частота взаимодействия с этим тегом будет 0.6. Это значит, что он в большинстве случаев выбирает именно эту тему.

Третья модель — модель вероятности ошибки в задании.

Она позволяет оценить риск того, что студент допустит ошибку при выполнении нового задания. Это особенно полезно, если мы хотим заранее определить, где студенту может понадобиться поддержка — например, дополнительные подсказки, обучающие материалы или консультация. Мы можем заранее определить, какие студенты скорее всего столкнутся с трудностями на конкретном задании, и превентивно предложить им помощь. Это делает процесс обучения **более адаптивным и персонализированным**, а также повышает общее качество образовательного процесса.