содержание

[Введение 4](#_Toc130461809)

[1 Основная часть 6](#_Toc130461810)

[1.1 Аппаратный сервер 6](#_Toc130461811)

[1.2 Настройка сервера 12](#_Toc130461812)

[Заключение 24](#_Toc130461813)

[Список литературы 25](#_Toc130461814)

# Введение

В настоящее время наблюдается непрерывное развитие информационных технологий в организациях разного рода деятельности и рост темпов появления новой информации. Это обусловлено стремлением к сокращению времени обработки и анализа данных. В связи с тем, что объем рабочей и иной информации является обширным, в любой организации появляется необходимость использования автоматизированной обработки данных, которая обеспечила бы наиболее эффективную работу. Большая часть организаций выбирают компьютеризированные способы, позволяющие эффективно хранить, структурировать и систематизировать большие объемы данных.

Одной из важнейших проблем качественной организации учебного процесса в высшем учебном заведении является задача создания автоматизированного учебного расписания. Многие люди считают, что опытные диспетчеры могут вручную составить расписание, оптимальное для учебного процесса и общественной жизни образовательного учреждения, но, учитывая увеличение количества студентов, укрупнение образовательных учреждений, необходимость использования и учёта большого количества данных, такое мнение, безусловно, можно считать ошибочным.

Потенциал вычислительной техники уже давно позволяет поставить и решить задачу создания автоматизированной системы составления расписания, которая упростила бы часть организации деятельности учебной части. Исследованию этой проблемы и ее разработке посвящены программы «1С:Автоматизированное составление расписания. Колледж», «Экспресс-расписание Колледж», «АВТОРасписание».

Однако образовательному учреждению довольно проблематично приобрести готовую программную продукцию, которую без труда можно было бы перенести на почву учебного заведения. Более реалистичной является подготовка частной системы, позволяющей планировать и составлять расписания занятий студентов и преподавателей.

# Основная часть

## Описание предметной области

Расписания можно использовать для организации и координации деятельности в различных условиях, таких как школа, бизнес и личная жизнь. Они могут помочь людям оставаться организованными и обеспечивать эффективное и своевременное выполнение задач.

Расписание - это план выполнения конкретной задачи или действия, часто в течение определенного периода времени. План включает в себя список задач или событий, которые необходимо выполнить, и время, в которое они должны произойти.

Существует несколько типов расписаний, которые можно использовать в зависимости от потребностей и целей человека или организации:

* годовой график - это план, в котором излагаются действия и задачи, которые необходимо выполнить в течение года. Он может включать ежемесячные расписания, а также любые повторяющиеся события или действия, которые происходят ежегодно;
* график проекта - это план, в котором излагаются задачи и этапы, которые необходимо выполнить в рамках конкретного проекта. Он включает график выполнения каждой задачи, а также может включать необходимые ресурсы, зависимости между задачами и крайние сроки;
* сменный график - это план, в котором указывается время, в которое должны работать разные сотрудники. Он может включать разные смены для разных дней недели, а также может включать выходные и праздничные дни;
* расписание занятий - это план, в котором указаны время и дни, в которые запланированы занятия для различных занятий или разделов курса. Он обычно используется в школах и университетах для организации времени и места занятий;
* расписание собраний - это план, в котором указаны время и место запланированных собраний. Это могут быть регулярные встречи, а также разовые или специальные встречи;
* график поездок - это план, в котором излагается маршрут поездки, включая время и место отправления/прибытия рейсов, гостиниц и других мероприятий, связанных с поездками;
* расписание событий - это план, в котором указаны сроки и детали события, например конференции, концерта или торговой выставки. Он может включать время и место проведения различных сессий или мероприятий, а также любые специальные меры или приспособления, которые необходимо сделать;
* график технического обслуживания - это план, в котором излагаются задачи и действия, которые необходимо выполнить для технического обслуживания и ремонта оборудования или помещений. Он может включать график выполнения этих задач, а также сведения о ресурсах и персонале, необходимых для их выполнения;
* производственный график - это план, в котором излагаются задачи и действия, связанные с производством товаров или услуг. Он может включать график выполнения каждой задачи, а также сведения о ресурсах и персонале, необходимых для их выполнения.

Расписание занятий может включать следующую информацию:

* название и номер курса;
* имя преподавателя;
* дни и время встречи;
* местонахождение, например, номер кабинета;
* название предмета.

Расписания занятий могут также включать сведения о формате занятия, например, очная форма обучения, онлайн или их сочетание. Расписание занятий обычно распространяется среди студентов в начале семестра и может быть изменено из-за непредвиденных обстоятельств или обновлений.

## Постановка задачи

Веб-приложение для составления расписания занятий «Advanced Schedule» будет основано на веб-сервере в операционной системе Linux.

Веб-приложение для составления расписания занятий «Advanced Schedule» будет брать на себя задачи по проверке расписания на пересечения и накладки. Также благодаря веб-интерфейсу приложения, оно будет доступно на любых устройствах с выходом в сеть интернет и с браузером.

Веб-приложение «Advanced Schedule» будет иметь следующие функции:

* обеспечение понятного для пользователя функционала;
* возможность просмотра и фильтрации информации в базе данных, например, группы, предметы, преподаватели (пользователи) и любые другие данные;
* добавление новой информации, составление расписания занятий, при этом система способна проверять расписание занятий на пересечение;
* просмотр расписания занятий любыми пользователями.

## Разработка алгоритма веб-приложения

Использование веб-приложения «Advanced Schedule» будет выполняться с двумя сценариями:

* всеобщий просмотр расписания обычными пользователями, такими как студенты, преподаватели, родители студентов;
* составление расписания ответственными лицами, администраторами веб-приложения.

Пользователь системы:

* выбирает режим просмотра расписания, присутствует три режима - просмотр по группам/аудиториям/преподавателям;
* после выбора режима, пользователь вводит текст для поиска необходимой(-ого) группы/аудитории/преподавателя;
* затем система автоматически проводит выборку из БД по тексту на предмет любых совпадений, не учитывая регистр текста;
* в случае, если выборка дала какой-либо результат - система выводит пользователю совпадения текста, в случае, если выборка не дала какой-либо результат - система выводит пользователю сообщение «Ничего не найдено»;
* пользователь выбирает из списка совпадений необходимую(-ого) группу/аудиторию/преподавателя;
* после выбора группы/аудитории/преподавателя пользователь выбирает дату;
* на основе данных, выбранных пользователем, система выполняет поиск расписания, затем, в случае, если выборка дала результаты, система выводит найденное расписание пользователю, в противном случае, ничего не выводится.

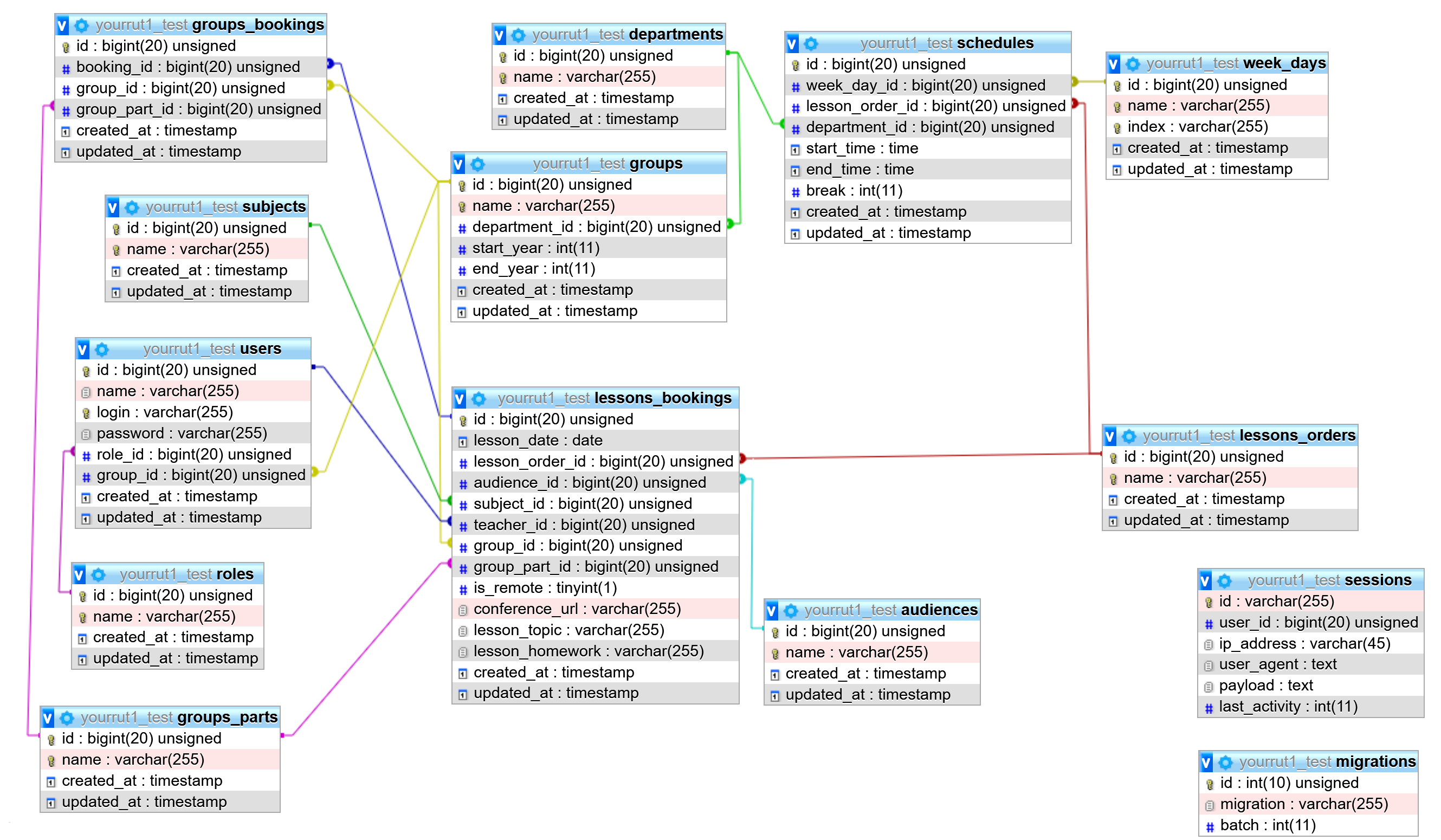
Администратор системы:

* проводит заполнение системы, вводя названия/ФИО необходимых групп/аудиторий/преподавателей, в случае, если эти данные еще не были введены в систему;
* после заполнения системы нужными данными, администратор составляет расписания, выбирая из списка необходимые параметры;
* затем система последовательно и автоматически проверяет корректность выбранных параметров, если при проверке была выявлена некорректность, то система сообщает об этом и администратор должен изменить набор параметров для составления расписания;
* после успешного процесса валидации данных, система вносит пару в БД и сообщает пользователю об успешной операции.

## Разработка схемы БД

База данных, являющаяся одной из важнейших частей информационной системы, конечно же, представляет собой сложный объект, который также подлежит проектированию.

В процессе создания информационной системы проектирование базы данных имеет очень важную роль, так как база данных является фундаментом информационной системы. Проектирование базы данных выполняется после анализа требований к будущей системе. А уже после того, как выработана общая схема базы данных, происходит процесс определения архитектуры будущей информационной системы. На рисунке 1 изображена схема спроектированной базы данных.



1. Схема БД

## Обоснование выбора языка программирования

В качестве языка программирования для разработки веб-приложения «Advanced Schedule» был выбран язык PHP.

PHP - C-подобный скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Язык программирования PHP спроектировал датский программист Расмус Лердорф в 1995 году как инструмент для создания интерактивных и динамических веб-сайтов. Сейчас PHP активно используют веб-разработчики для создания сайтов и веб-приложений. Это наименее конкурентный язык для работы. Его использует большинство сервисов, и, скорее всего, ситуация не изменится в ближайшие 10 лет. Поэтому специалисты очень востребованы.

Основная область применения - разработка скриптов, которые работают на стороне сервера. Также можно создавать скрипты командной строки и GUI-приложения.

PHP используется в проектах разных масштабов: от браузерной игры и социальной сети до крупных веб-приложений в интернете, таких как Facebook, ВКонтакте, Google, W3C, Yahoo, NASA. На PHP написаны многие системы управления контентом, например, Drupal, WordPress, Joomla. PHP использовался при разработке популярных фреймворков для создания сайтов, например, Yii2, Laravel, Symfony.

PHP - язык с динамической типизацией. Это означает, что переменная не определяется жестко и заранее. Динамическая типизация позволяет присваивать одной и той же переменной несколько значений: она может быть массивом, строкой, числом, объектом. Динамическая типизация удобная и гибкая, но приводит к потреблению большого количества оперативной памяти и уменьшает скорость работы.

PHP - интерпретируемый язык. Это означает, что код, который написан на PHP, не компилируется, а выполняется строка за строкой с помощью программы-интерпретатора, которая читает каждую из них и выполняет то, что там указано.

PHP хранит состояние, созданное в текущий момент времени. В отличие от классических компилируемых языков программирования, которые работают все время от запуска до остановки программы, PHP-скрипт начинает выполняться с момента обращения веб-сервера и заканчивает, когда возвращает сгенерированный пакет данных обратно веб-серверу.

Достоинства:

* для PHP есть множество баз данных, фреймворков и библиотек;
* язык бесплатный, код находится в открытом доступе;
* язык гибкий и предоставляет много свободы;
* PHP совместим практически со всеми современными серверами;
* простой синтаксис;
* PHP подходит новичкам, ему легко научиться и практиковать полученные знания можно почти сразу;
* язык имеет простую систему объектов и классов.

Недостатки:

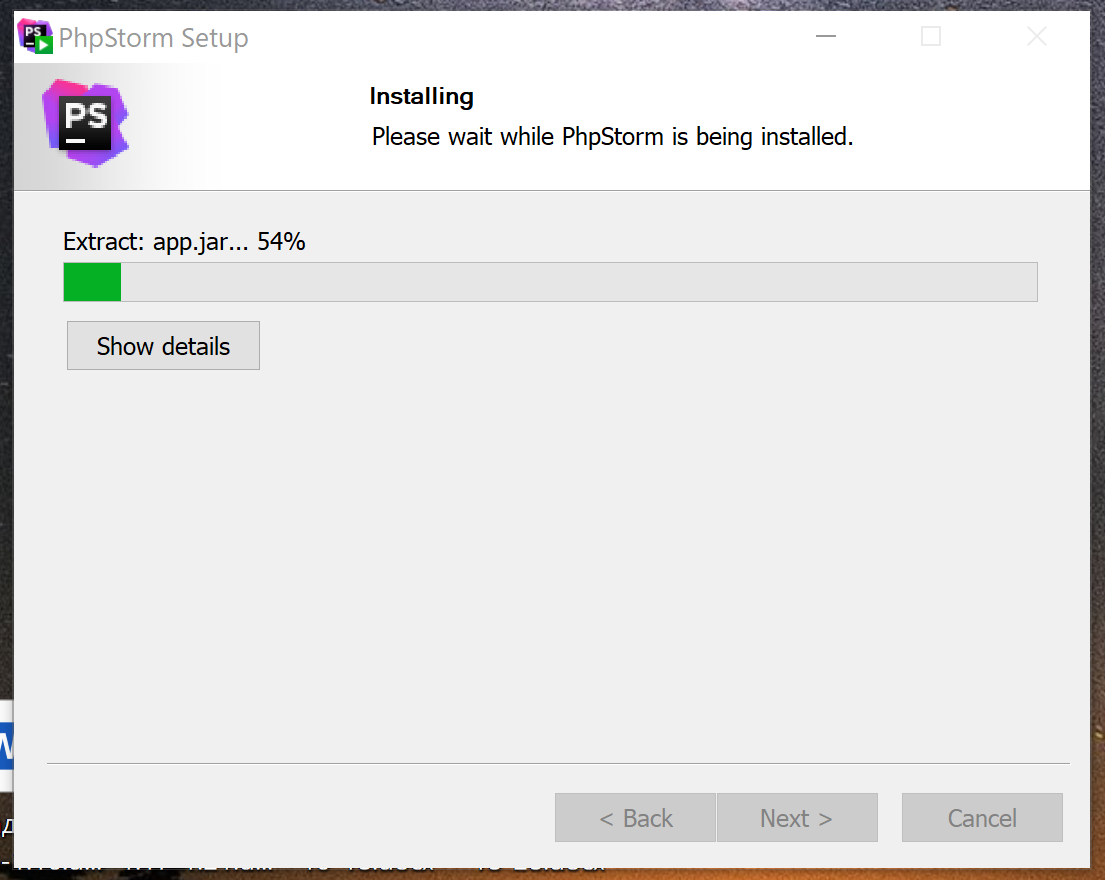
* для работы с PHP желательно знание HTML и CSS;
* из-за простоты и гибкости PHP сложно поддерживать. Написать плохой код очень легко, найти ошибку тяжело;
* PHP — не очень производительный язык. Его конкурент JavaScript быстрее;
* из-за смеси PHP и HTML на многостраничных сайтах может быть затруднительно найти необходимую часть кода;
* в названиях функций стандартной библиотеки отсутствует четкая система: одни имеют сокращения и подчеркивания, другие — нет. Иногда в названиях функций для работы со строками встречаются обозначения str, а иногда их не бывает.

## Описание процесса разработки веб-приложения

Исходя из поставленной задачи были проведены следующие работы:

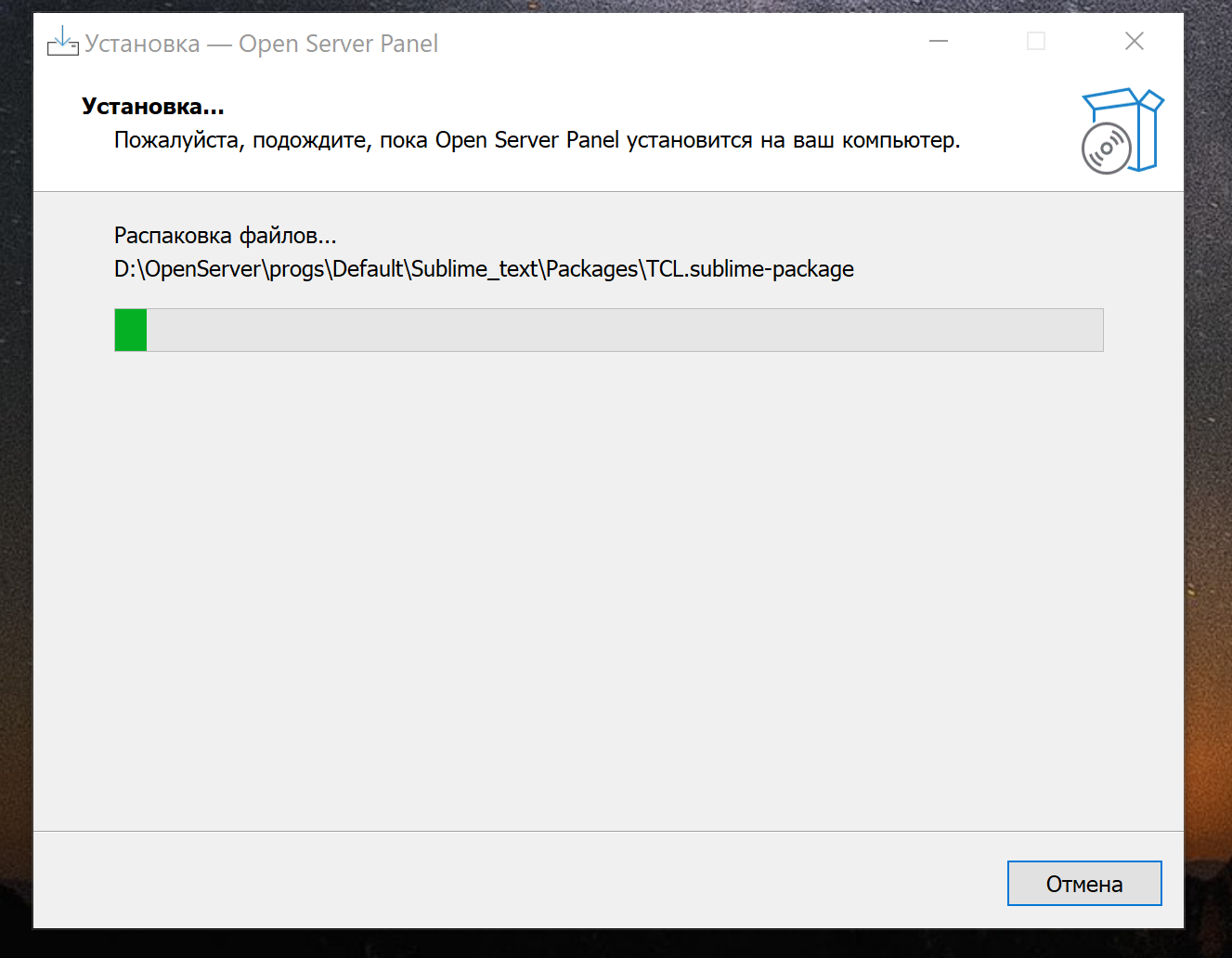
* загрузка и установка интегрированной среды разработки JetBrains PhpStorm;
* настройка интегрированной среды разработки JetBrains PhpStorm;
* загрузка и установка программы Open Server, реализующей функции для локальной разработки веб-приложений, а именно: веб-сервер, сервер базы данных, интерпретатор языка программирования PHP;
* настройка модулей Open Server, выбор версии языка программирования PHP, выбор веб-сервера, выбор модуля СУБД;
* создание нового проекта;
* настройка зависимостей и установка программного обеспечения для разработки приложения с помощью терминального пакетного менеджера «composer»;
* конфигурирование проекта и фреймворка Laravel;
* настройка зависимостей и установка фреймворка Vue JS, вспомогательной библиотеки Vuex с помощью пакетных менеджеров «composer» и «npm»;
* разработка программного кода;
* тестирование и отладка приложения.

Для загрузки интегрированной среды разработки JetBrains PhpStorm необходимо перейти на официальный сайт разработчиков, перейти в раздел загрузки и начать скачивание. Важно, что разработчики распространяют свою среду разработки на платной основе. Имеется 30-дневный пробный период, чего вполне достаточно для разработки. После скачивания установочной программы с расширением «exe» был произведен запуск установки интегрированной среды разработки JetBrains PhpStorm. На рисунке 2 изображен процесс настройки комплектации среды разработки и установки.



1. Интерфейс окна установки JetBrains PhpStorm

Для разработки и тестирования веб-приложения требуется интерпретатор языка программирования PHP, веб сервер и сервер базы данных. Программа Open Server для локальной разработки веб-приложений предоставляет все эти функции из коробки. На рисунке 3 представлен процесс установки Open Server.

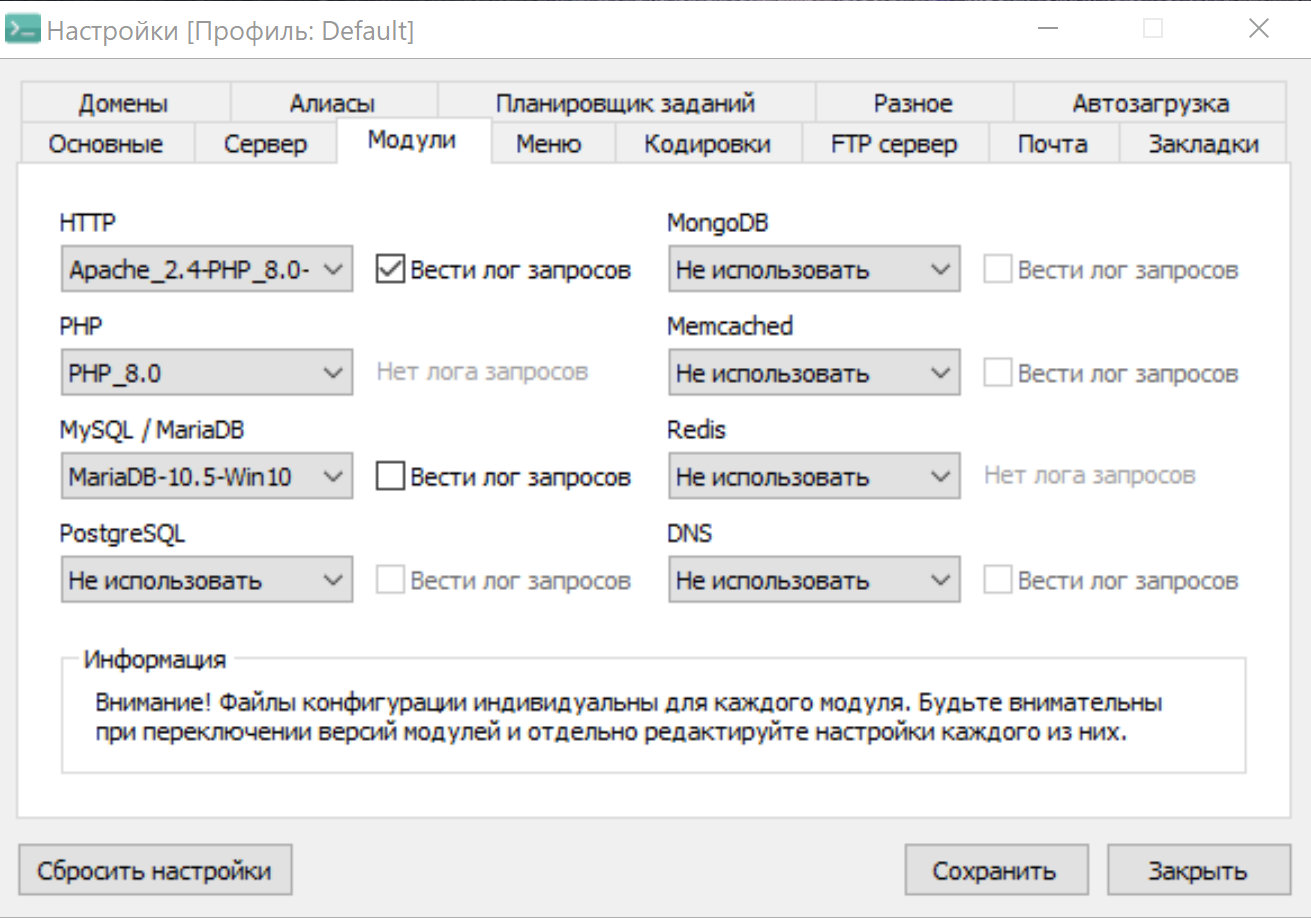


1. Интерфейс окна установки Open Server

Программа Open Server включает в себя множество модулей веб-серверов, серверов баз данных разных версий. Поэтому требуется настройка требуемых версий модулей для разработки веб-приложения «Advanced Schedule», а именно:

* веб-сервер Apache версии 2.4;
* интерпретатор языка PHP версии 8.0;
* сервер реляционной базы данных MariaDB версии 10.5.

На рисунке 4 изображено окно настройки Open Server.

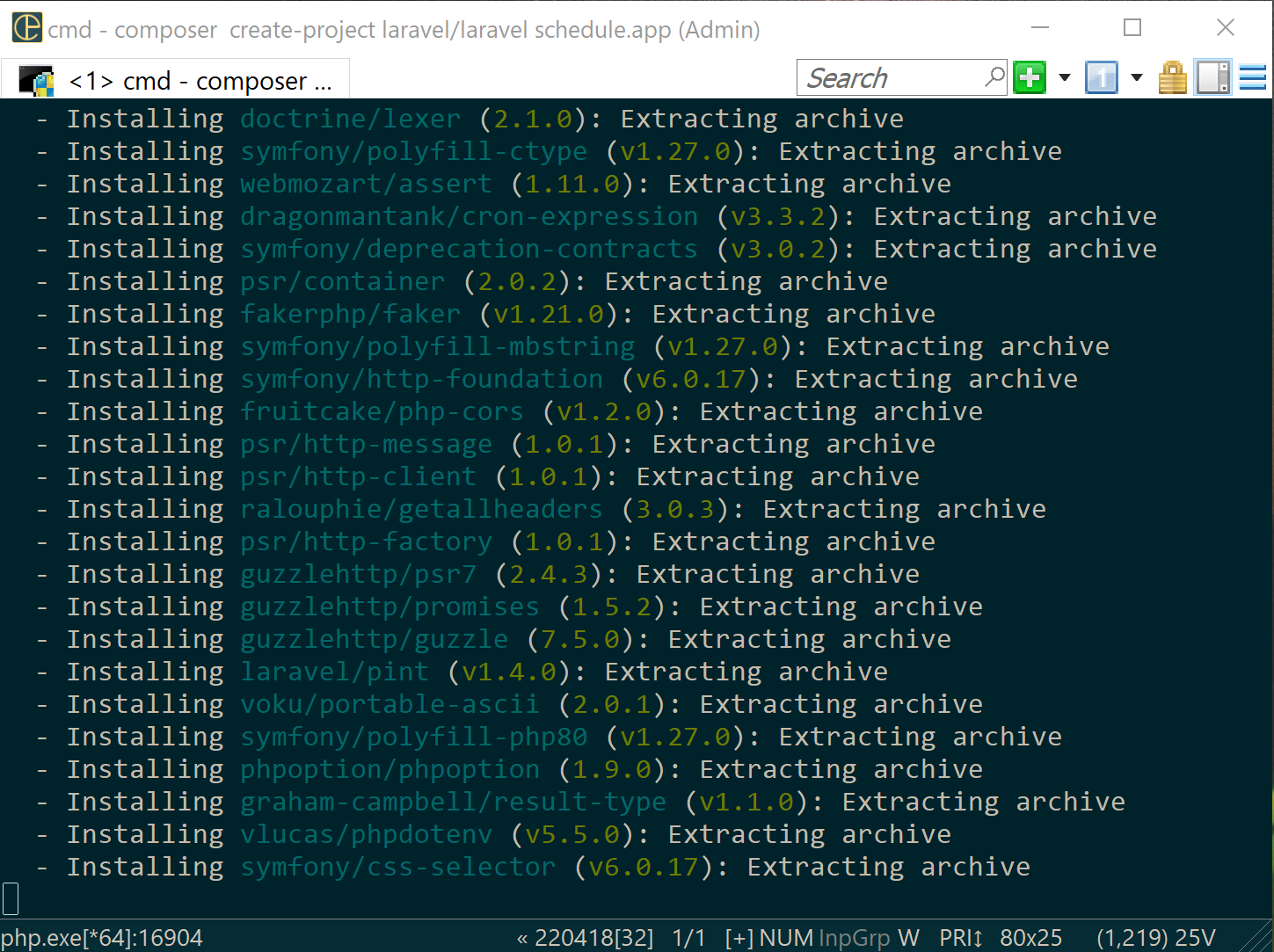


1. Интерфейс окна настройки Open Server

После завершения настройки Open Server необходимо создать проект в директории «domains» данной программы с помощью пакетного менеджера Composer, выполнив следующие шаги:

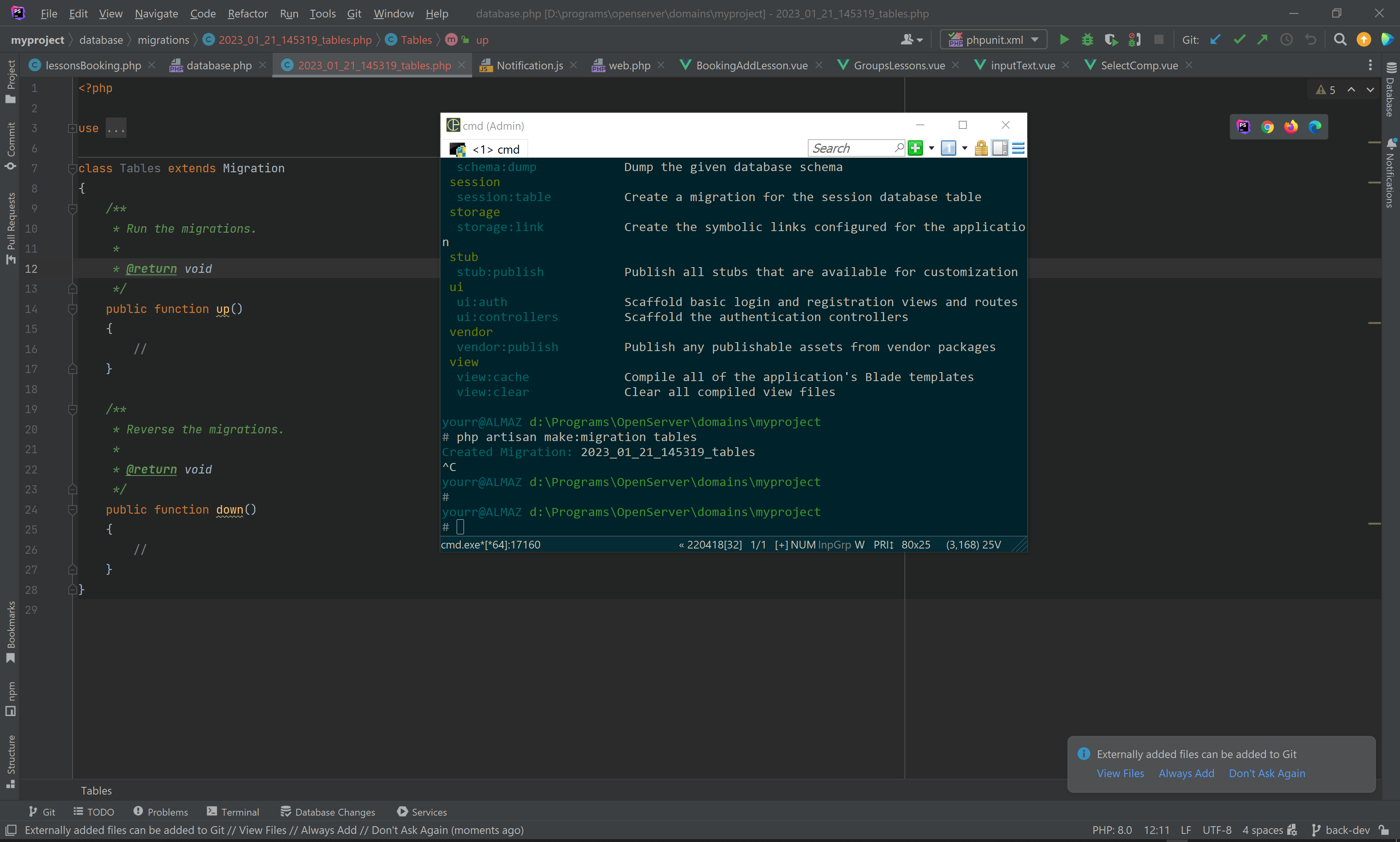
* запустить Open Server и открыть консоль поставляемую вместе с данной программой;
* в консоли перейти в директорию «domains»;
* ввести команду «composer create-project laravel/laravel schedule.app», где «laravel/laravel» обозначает создать пустой проект с фреймворком Laravel, а «schedule.app» название директории, в которую будет помещен созданный проект.

На рисунке 5 показан интерфейс окна терминала во время процесса создания нового проекта.



1. Интерфейс окна терминала во время процесса создания нового проекта и загрузки зависимостей с помощью пакетного менеджера Composer

Первым этапом в разработке программного кода веб-приложения является составление структуры базы данных. Во фреймворке Laravel это осуществляется с помощью инструмента миграций. В файлах миграций задаётся структура таблиц и их связи. Для того, чтобы создать файл миграции, фреймворк Laravel предоставляет специальный помощник в терминале – «artisan»; необходимо обратиться к нему с соответствующей командой. На рисунке 6 изображен интерфейс окна терминала после создания файла миграции и как выглядит программный код данного файла миграции.



1. Интерфейс окна терминала после создания файла миграции

В файле миграции присутствуют две функции:

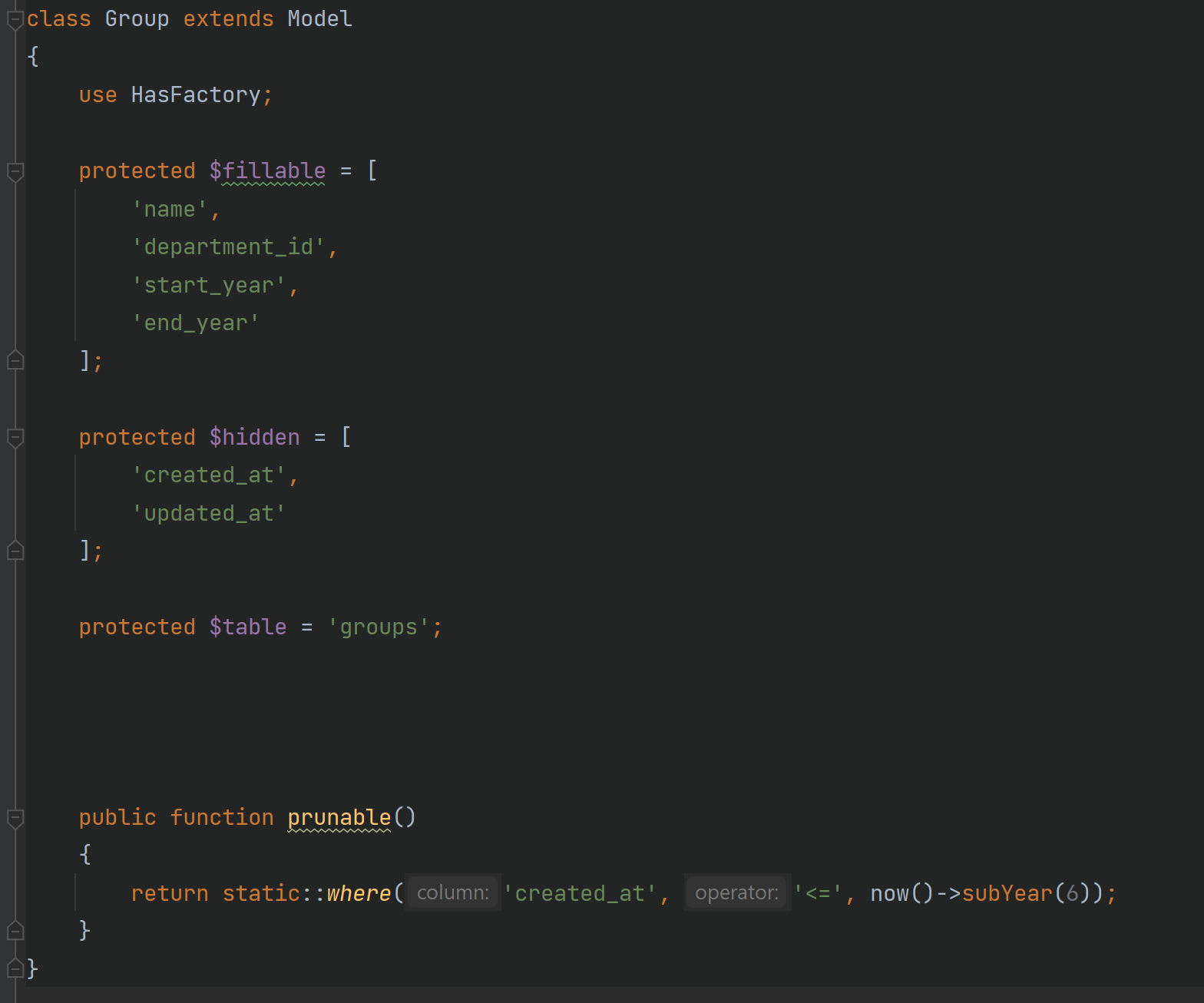
* up() – отвечает за действия, которые будут совершаться при выполнении миграции, это может быть создание таблиц;
* down() – отвечает за действия, которые будут совершаться при отмене миграций, это может быть удаление созданных таблиц при выполнении миграций.

На рисунке 7 отображено каким образом заполняется файл миграции.



1. Программный код файла миграции

После того, как структура базы данных задана, необходимо описать модели для работы с каждой таблицей, определить в них необходимые функции, которые позволят взаимодействовать с данными. На рисунке 8 изображен программный код модели.



1. Программный код модели

Модели для работы с базой данных определены. Для того, чтобы запросы проходили через программный код и попадали в нужные методы и в нужные контроллеры, необходимо определить маршруты запросов по программному коду. За маршруты отвечают специальные файлы для маршрутизации, с английского языка «router». Фреймворк Laravel предоставляет несколько маршрутизаторов для различных целей. В случае веб-приложения будет использован соответствующий роутер «web». На рисунке 9 представлен программный код маршрутизатора и заданные маршруты запросов.



1. Программный код маршрутизатора запросов

# Заключение

В ходе прохождения преддипломной практики были продемонстрированы теоретические и практические навыки и умения, полученные в рамках обучения. Подводя ее итоги можно выделить следующие наиболее значимые достигнутые результаты:

* проведен анализ существующей организации информационных процессов составления расписания в колледже и выявлены недостатки, устранение которых возможно за счет внедрения автоматизированной системы составления расписания;
* спроектированы база данных и автоматизированная система составления расписания в колледже;
* разработано программное обеспечение, внедрение которого позволяет обеспечить повышение эффективности и снижение трудозатрат персонала ГБПОУ Уфимского государственного колледжа технологии и дизайна посредством автоматизации информационных процессов составления расписания в школе.

Разработанная система реализована средствами современных сетевых web-технологий с целью автоматизации организационных процессов и обеспечения возможности одновременного отдаленного доступа пользователей к информационным ресурсам.

В результате проделанной работы была разработана автоматизированная система расписания занятий образовательного заведения с целью ее внедрения в учебный процесс образовательного учреждения. Данная система имеет удобный пользовательский интерфейс, позволяющий легко освоить работу в программе. Преследуемая изначально цель автоматизации процесса расписания занятий достигнута. Разработанная система позволит повысить скорость обработки информации, сократит сроки формирования расписания и сэкономит время работы пользователей.

# Список литературы

1. Автоматизация составления расписания учебных занятий в вузе // Материалы Всероссийской научно-технической конференции «Наукоемкие технологии в приборо- и машиностроении и развитие инновационной деятельности в ВУЗе». - Т.2. - М., Издательство МГТУ им. Н.Э.Баумана, 2008. - С. 98.
2. Атрощенко В.А. К вопросу проектирования автоматизированной системы составления расписаний с учетом приоритетов заявок // Сборник международной научно- практической конференции «Научные исследования и их практическое применение. Современное состояние и пути развития 2010». Том 5. / Атрощенко В.А., Семенюта И.С. - Одесса: Черноморье, 2010. - С. 55-58.
3. Батищев, П.С. Опыт использования информационных технологий при составлении расписания учебных занятий. Текст. // Среднее профессиональное образование. - №11. - 2003.
4. Беленький, A.C. Применение моделей и методов теории расписаний в задачах оптимального планирования на грузовом транспорте. Текст. / A.C. Беленький, Е.В. Левнер // Автоматика и телемеханика. 1989. -№1.- С. 3-77.А
5. Брезгинов, А.Н. Обзор существующих методов составления расписаний / А.Н. Безгинов, С.Ю. Трегубов // Информационные технологии в программировании. - М., 2005. - №2(14).
6. Воробович, О.Н. Алгоритм формирования расписания занятий студенческих групп в высшем учебном заведении Текст. / О.Н. Воробович // Материалы межвузовской научной конференции / под ред. Е.А. Вейсова, Ю.А. Шитова, КГТУ. - Красноярск, 2003.- С.29-35.
7. Воробович, О.Н. Информационная система формирования расписания занятий в высшем учебном заведении Текст. / О.Н. Воробович // Вестник СибГТУ, N1 / СибГТУ. Красноярск 2003. - С. 120-125.