Титульный лист

МИНИСТЕРСТВООБРАЗОВАНИЯМОСКОВСКОЙ ОБЛАСТИ

Государственное бюджетное профессиональное образовательное учреждение Московской области«Ногинский колледж»

Курсовой проект

по МДК.09.01Проектирование и разработка веб-приложений

ПМ.09 Проектирование, разработка и оптимизация веб-приложений

Тема:

ИСПОЛЬЗОВАНИЕ ФРЕЙМВОРКА LARAVEL В РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЯ «СИСТЕМА УПРАВЛЕНИЯ ПЕРСОНАЛОМ И РАБОЧИМИ ПРОЦЕССАМИ»

Разработчик: студент группы 3ИСПР1

Бойко Д.Ю

(подпись)

Оценка защиты курсового проекта

« »

Дата защиты

« » 2024

Руководитель проекта:

преподавательСтепанов С.О.

(подпись)

Ногинск,2024г   
Введение

Целью данного курсового проекта была разработка веб-приложения на тему «Система управления персоналом и рабочими процессами»

Мною были поставлены такие задачи, как:

1. Разработка структуры и функционала сайта.
2. Настройка базы данных.
3. Вёрстка сайта.
4. Тестирование и отладка функционала.

Современные веб-сайты, посвященные системам управления персоналом и рабочими процессами, предлагают широкий спектр информации, инструментов и решений для организаций любого масштаба. Они помогают компаниям повысить производительность, улучшить коммуникацию внутри коллектива, автоматизировать рутинные задачи и улучшить управление ресурсами.

Основная часть

Общие сведения о разработке веб-приложения на фреймворке

Laravel и Bootstrap

Laravel — это PHP-фреймворк, предназначенный для разработки современных веб-приложений. Он следует архитектуре MVC (Model-View-Controller) и способствует разработке чистого и структурированного кода. Laravel обладает множеством качеств и функций, которые делают его привлекательным для разработчиков:

1. MVC Архитектура: Позволяет разделять логику приложения (Model), пользовательский интерфейс (View) и управление данными (Controller).
2. Eloquent ORM: Предоставляет простой и интуитивно понятный способ взаимодействия с базой данных.
3. Blade Шаблонизатор: Легкий и мощный механизм для создания шаблонов с переиспользуемыми компонентами.
4. Аутентификация и Авторизация: Встроенные механизмы для управления пользователями и правами доступа.
5. Маршрутизация: Гибкая система маршрутизации, поддерживающая как простые, так и сложные маршруты.
6. Расширяемость: Большое количество готовых пакетов и возможность создания собственных расширений.

Bootstrap — это популярный фронтенд-фреймворк, разработанный для создания адаптивных и удобных в использовании интерфейсов. Он предоставляет обширный набор предопределенных стилей и компонентов, что позволяет создавать привлекательные и функциональные решения с минимальными усилиями.

Готовые компоненты: Bootstrap предоставляет множество готовых компонентов, таких как кнопки, формы, навигационные панели, карточки, шаблоны и примеры, которые можно использовать для быстрого создания интерфейсов. Эти компоненты стандартизированы и протестированы, что гарантирует их корректную работу во всех современных браузерах.

Адаптивность: Bootstrap использует сеточную систему и медиазапросы для обеспечения корректного отображения страниц на различных устройствах — от мобильных телефонов до настольных компьютеров. Адаптивность позволяет создавать интерфейсы, которые автоматически подстраиваются под размер экрана пользователя, обеспечивая удобство использования на любых устройствах.

Использование Laravel и Bootstrap в одном проекте предоставляет мощный набор инструментов для разработки веб-приложений. Laravel обеспечивает мощный и многофункциональный сервер, предоставляя всю необходимую серверную логику и управление данными, в то время как Bootstrap предоставляет стильный и адаптивный интерфейс для фронтенд-разработки.

Laravel и Bootstrap представляют собой мощный тандем для разработки веб-приложений. Laravel предоставляет все необходимые инструменты для создания серверной части приложения, обеспечивая удобную работу с базами данных, маршрутизацией, аутентификацией и другими важными аспектами. Bootstrap, в свою очередь, предлагает обширный набор готовых компонентов и стилей для создания адаптивного и современного интерфейса. В результате, использование этих фреймворков позволяет создавать функциональные, привлекательные и поддерживаемые веб-приложения с минимальными усилиями.

**Установка необходимых программ и фреймворка**

##### 1. Установка PHPStorm

PHPStorm является интегрированной средой разработки, которая упрощает процесс написания и редактирования кода благодаря функциям автодополнения, подсветки синтаксиса и генерации кода.

##### 2. Установка Composer

Composer — это инструмент для управления зависимостями в PHP, который позволяет вам объявлять библиотеки, от которых зависит ваш проект, и управлять ими.

##### 3. Установка OpenServer

OpenServer — это портативная серверная платформа и программное обеспечение для веб-разработки, которое работает на Windows.

##### 4. Создание репозитория на GitHub

GitHub — это веб-сервис для управления проектами и репозиториями кода, использующий систему контроля версий Git.

##### 5. Установка фреймворка Laravel

Выполняется командой composer create-project --prefer-dist laravel/laravel your-project-name.

6.После установки проекта , нам надо настроить файл .env. Он отвечает за подключение сайта и базы данных а также он является одни из главных файлов запуска веб-приложения.

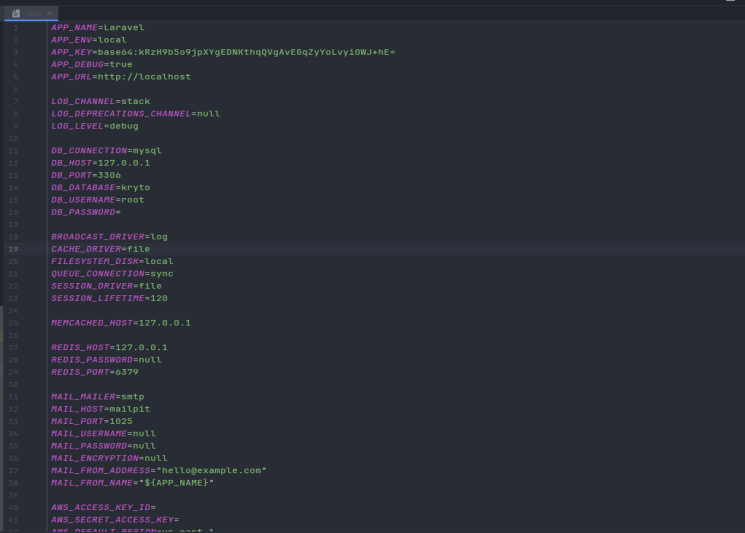


Рис.1.1

7.Так же мы можем настройки файл корневого каталога .htaccess

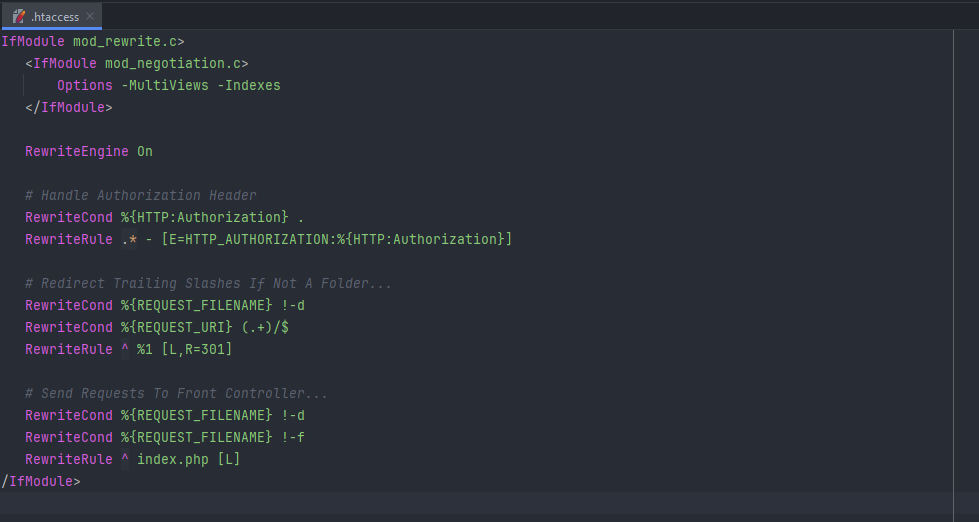


Рис.1.2

8.После настройки файла мы открываем консоль и заходим в папку domains , а потом в папку с проектом.

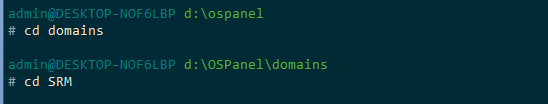


Рис.1.3

9.После того как мы зашли в папку мы можем прописать команду которой мы будем пользоваться всю работу , чтобы она запускала нам приложение .

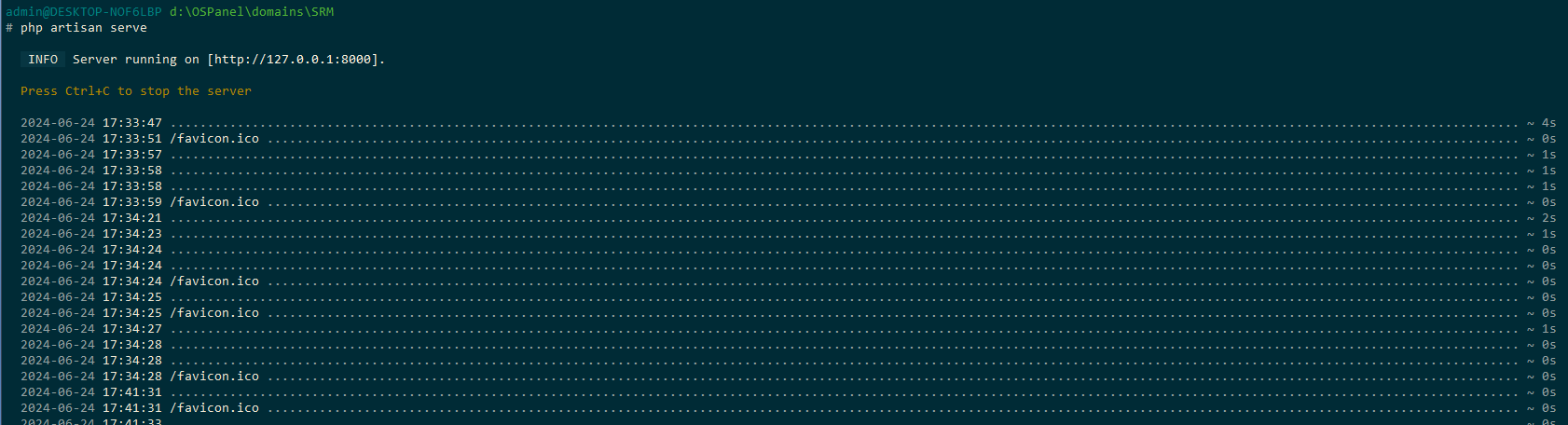


Рис.1.4

10. Открываем браузер и переходим по адресу http://127.0.0.1:8000, чтобы убедиться, что проект успешно запущен.

**Руководство** **программиста**

Laravel – это мощный и гибкий фреймворк для разработки веб-приложений на PHP, который обеспечивает быструю разработку благодаря готовым компонентам и инструментам. Он позволяет создавать как простые веб-сайты, так и сложные корпоративные приложения, предоставляя разработчикам удобные инструменты для реализации MVC-архитектуры, безопасности, кэширования, работы с базами данных и формами. Laravel также поддерживает пакеты и расширения, которые могут добавлять новые функции или интегрировать сторонние библиотеки, делая его одним из самых масштабируемых и модифицируемых фреймворков на рынке.

### Основная структура проекта

При первоначальном запуске проекта Laravel автоматически создается множество папок, которые работают совместно для обеспечения функциональности и организации кода. Вот краткое описание каждой из этих папок:

**.git -** содержит все необходимые файлы для управления версиями с помощью системы контроля версий Git. Она скрыта и не требует ручного вмешательства. Эти файлы управляют историей изменений и облегчают совместную работу над проектом.

**.idea -** используется интегрированной средой разработки (IDE), такой как PhpStorm, для хранения настроек проекта. Она может включать файлы конфигурации, такие как параметры кодировки, версии PHP и другие специфические настройки IDE, что помогает поддерживать единообразие среды разработки между участниками команды.

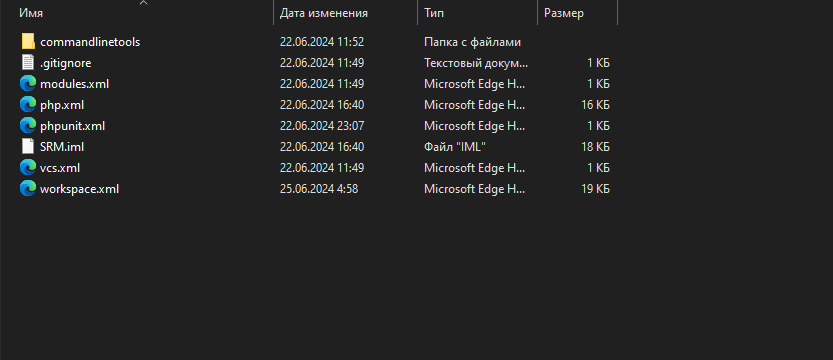


Рис.2.1

app:

Эта папка содержит различные подкатегории и файлы, которые структурируют логику и функциональность приложения. Вот подробное описание того, что можно найти в этой папке:

Папка Console предназначена для хранения всех консольных команд, которые могут быть выполнены через Artisan. Здесь можно создавать и регистрировать новые команды, расширяя функциональность приложения.

**Console/Commands:** Содержит файлы команд, которые выполняются через Artisan. Позволяет разработчикам создавать собственные команды для выполнения различных задач.

**Kernel.php:** Определяет расписание выполнения команд и регистрирует команды, обеспечивая автоматическое выполнение задач по расписанию.

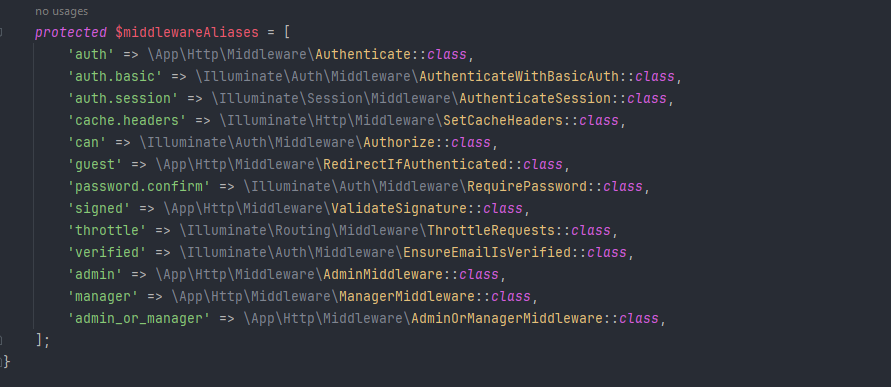
****

Рис.2.2

Папка Exceptions обрабатывает все исключения и ошибки, возникающие в приложении.

**Handler.php:** Основной файл для обработки исключений. Настраивает и управляет логикой обработки исключений, возникающих в приложении, и отвечает за их логирование.

### Http

**Назначение:** Папка Http содержит основные компоненты для обработки HTTP-запросов, включая контроллеры, посредники (middleware) и классы запросов**.**

**Controllers:** Содержит контроллеры, отвечающие за обработку HTTP-запросов. Контроллеры управляют взаимодействием между моделями и представлениями, реализуя логику приложения.

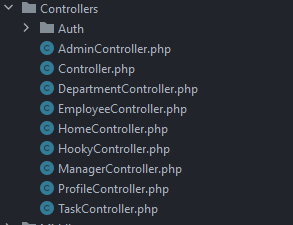


Рис.2.3

**Middleware:** Содержит файлы посредников, которые фильтруют и обрабатывают HTTP-запросы, проходящие через приложение. Посредники выполняют такие задачи, как аутентификация и проверка прав доступа.

**Requests:** Содержит классы запросов для валидации данных, поступающих в приложение. Эти классы помогают централизовать и упростить процесс валидации входных данных**.**

### Models

**Назначение:** Папка Models хранит модели данных, которые взаимодействуют с базой данных.

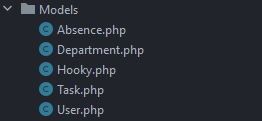
****

Рис.2.4

**User.php:** Модель пользователя, которая взаимодействует с таблицей users в базе данных. Модель содержит атрибуты и методы, представляющие сущности пользователя в приложении, и может быть расширена методами и свойствами, специфичными для логики приложения.

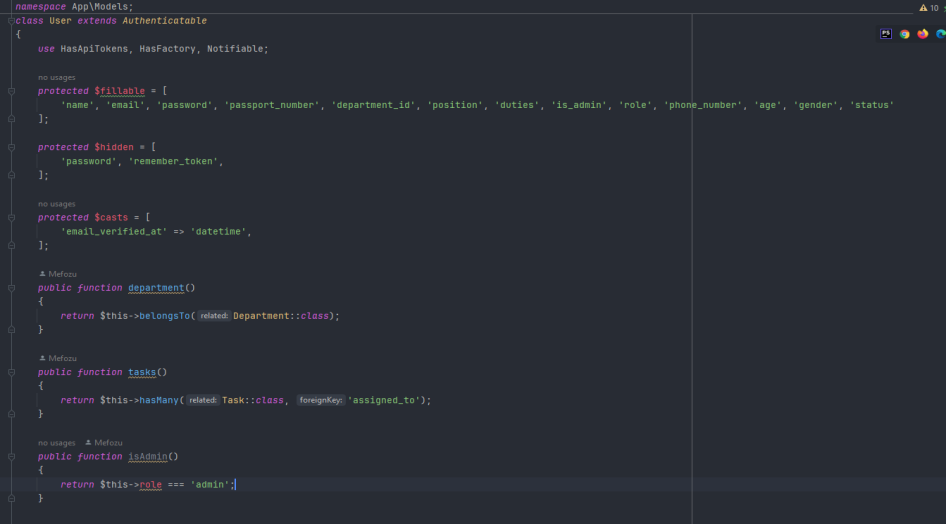


Рис.2.5

В данном коде мы прописываем роли которые у нас будут выводится на сайте и вносим строки которые у нас будут в дальнейшем выводится.

### Providers

**Назначение:** Папка Providers содержит классы поставщиков сервисов (Service Providers), которые инициализируют и конфигурируют основные компоненты приложения.

**Providers:** Классы поставщиков сервисов регистрируют и загружают различные службы и компоненты, такие как маршрутизация, события и другие службы, необходимые для работы приложения.

Папка Views обычно располагается в каталоге resources, однако в нашем случае она также присутствует в app. Эта папка содержит шаблоны представлений нашего приложения. Представления используются для отображения данных пользователям и являются важной частью MVC-архитектуры. Файлы представлений обычно имеют расширение .blade.php и используют шаблонизатор Blade, который предлагает удобные инструменты для построения HTML-страниц с динамическим контентом.

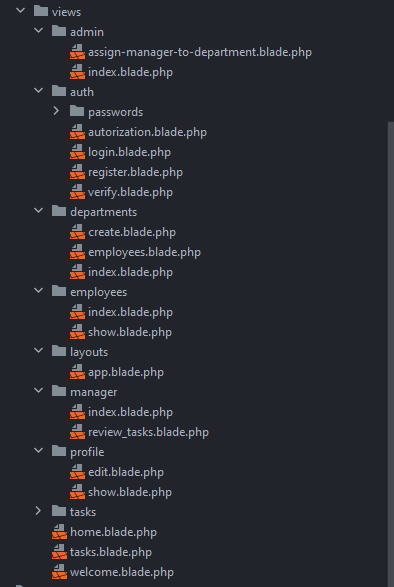


Рис.2.6

### Дополнительные файлы и их функции

**User.php:** Как упоминалось ранее, этот файл представляет собой модель пользователя. Она использует ORM Eloquent для взаимодействия с базой данных. Модель определяет атрибуты и методы, которые позволяют манипулировать данными пользователей, такими как создание, обновление и удаление записей.

Взаимодействие между папками app и resources происходит через MVC (Model-View-Controller) архитектуру. Папка app отвечает за бизнес-логику, обработку данных и контроллеры, а папка resources отвечает за представления и статические ресурсы.

#### Пример взаимодействия между app и resources

**HTTP-запрос пользователя:** Пользователь отправляет HTTP-запрос, например, при посещении URL /profile.

**Маршрутизация:** В Laravel маршруты обычно определяются в файле routes/web.php. Запрос пользователя сопоставляется с определенным маршрутом, который указывает на нужный контроллер и метод.

Config

Папка config в Laravel содержит файлы конфигурации, которые управляют различными аспектами работы приложения. Эти файлы позволяют настроить базовые параметры системы, интеграцию с внешними сервисами, параметры безопасности и многое другое. Каждое приложение Laravel поставляется с набором стандартных файлов конфигурации, которые можно изменять в соответствии с требованиями проекта.

#### Основные файлы и их назначение:

**app.php-** Включает настройки имени приложения, среды (environment), ключа шифрования, провайдеров служб и алиасов классов.

**auth.php-** Определяет guard'ы и провайдеры для аутентификации пользователей, а также настройки для сброса пароля.

**broadcasting.php-** Определяет драйверы вещания (например, Pusher, Redis) и их параметры.

**cache.php- Параметры:** Определяет драйверы кэша (файл, база данных, Redis) и их конфигурацию.

**database.php- Параметры:** Определяет параметры соединения для различных баз данных (MySQL, PostgreSQL, SQLite, SQL Server).

**filesystems.php-** Определяет драйверы файловых систем (локальные, FTP, S3) и их конфигурацию.

**logging.php-** Определяет каналы логирования (стек, один файл, дневник) и их параметры.

**mail.php-** Определяет драйверы отправки почты (SMTP, Mailgun, SES) и их конфигурацию.

**queue.php- Параметры:** Определяет драйверы очередей (sync, database, Redis, Beanstalkd) и их параметры.

**services.php- Параметры:** Определяет параметры для интеграции с внешними API (Stripe, Mailgun, SES).

**session.php-** Определяет драйверы сессий (файловая система, cookie, база данных, Redis) и их конфигурацию.

**view.php-** Определяет пути к шаблонам и кэшированию представлений.

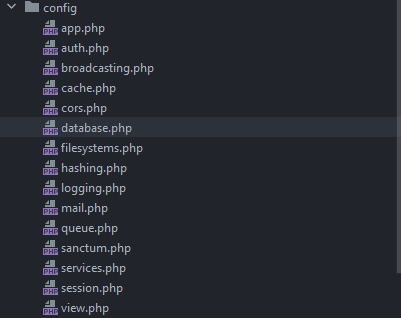


Рис.2.7

Database

Папка database предназначена для работы с базой данных. Она содержит миграции, фабрики и сидеры, которые помогают управлять схемой базы данных, создавать тестовые данные и инициализировать базу данных с начальными данными.

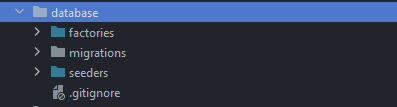


Рис.2.8

Папка factories - содержит фабрики для генерации тестовых данных. Фабрики используются для создания моделей с заполнением данных по умолчанию.

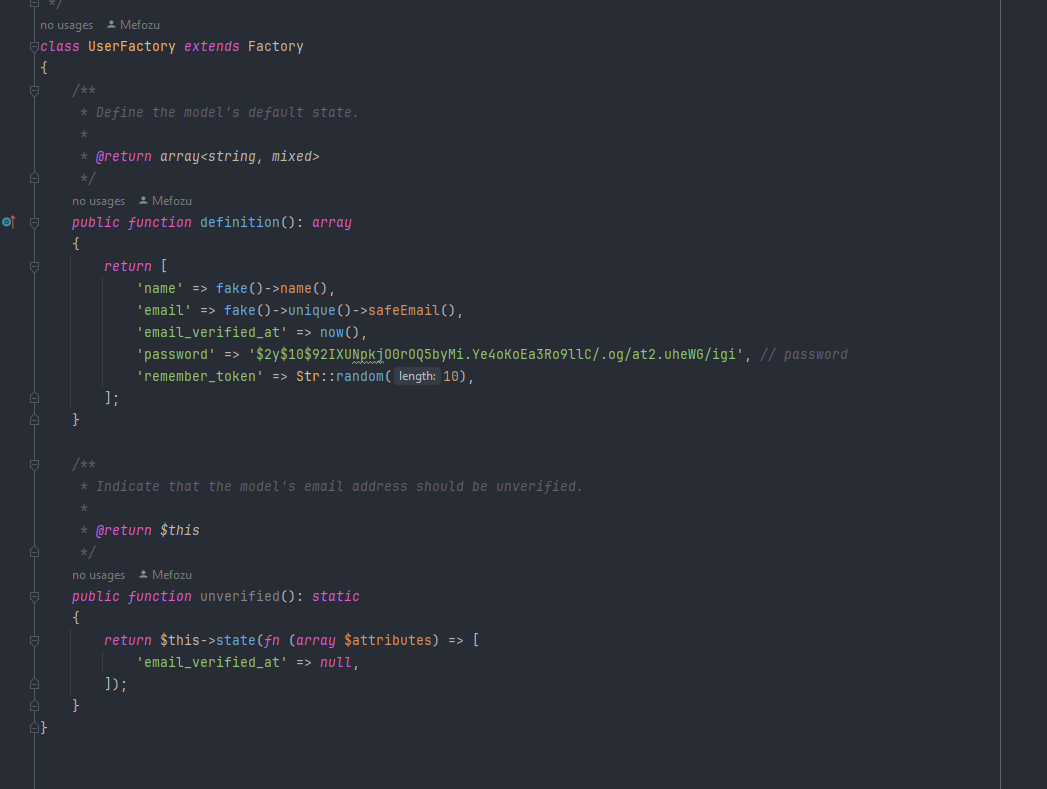


Рис.2.9

Папка migration – cодержит файлы миграций, которые описывают изменения в структуре базы данных. Миграции позволяют версионировать базу данных и обеспечивают возможность отката изменений.

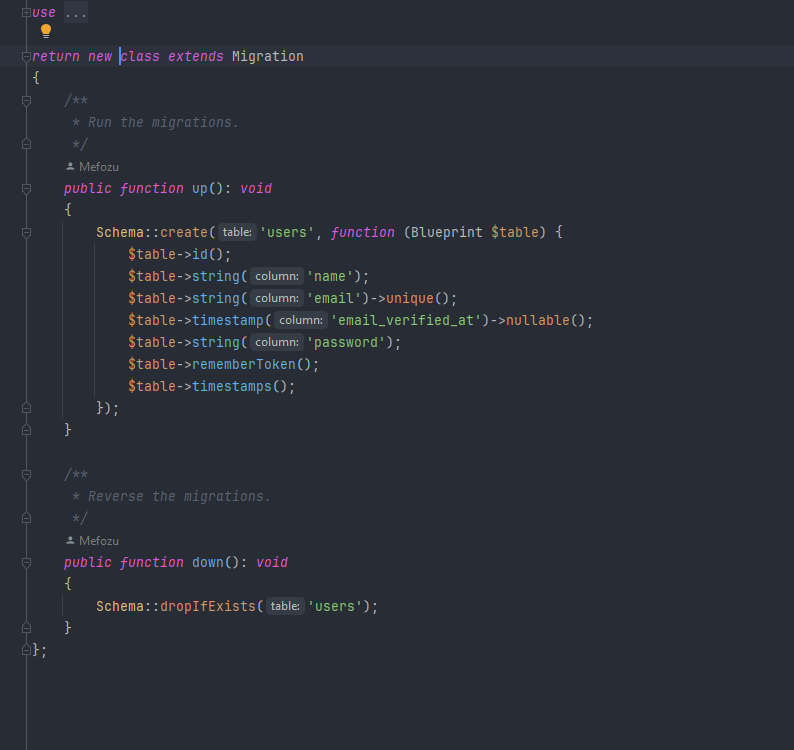


Рис.2.10

Папка public - выполняет роль корневой директории веб-сервера. Она содержит файлы, которые доступны для публичного доступа и являются точкой входа для всех запросов к вашему приложению.

Папка resources является одной из главной папки в проекте , так как в нее мы всегда вписываем свой дизайн , таблицы , поля для ввода и тд.

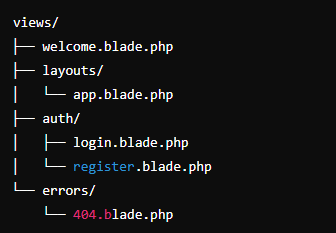


Рис.3.1

2.Установка веб-приложения на общедоступный хостинг

3.Заключение

В ходе данной работы у меня получилось создать свой проект "Система управления персоналом и рабочими процессами", используя фреймворки Laravel и Bootstrap. Этот процесс включал несколько ключевых этапов, начиная от разработки и настройки проекта на локальном сервере до его успешного развертывания на хостинге.

Laravel обеспечил мощную серверную часть, предлагая удобные инструменты для работы с базой данных, маршрутизацией и аутентификацией пользователей. Bootstrap позволил создать удобный пользовательский интерфейс, используя предопределенные стили и компоненты, что значительно ускорило процесс разработки фронтенда.

Поставленные задачи были выполнены. Я разработал и сверстал систему с работающим функционалом управления задачами, отслеживания отсутствий сотрудников, а также с удобным интерфейсом для администраторов и менеджеров.

Работа над проектом "Система управления персоналом и рабочими процессами" позволила мне не только ознакомиться с возможностями фреймворков Laravel и Bootstrap, но и углубиться в процесс разработки и развертывания веб-приложений. Благодаря структурированному подходу и детальному рассмотрению каждого этапа, мне удалось создать функциональное и стабильное приложение, готовое к использованию пользователями.

Этот проект продемонстрировал важность внимательного планирования и выполнения каждого шага, начиная от начальной настройки среды разработки и заканчивая финальными проверками. Понимание внутренней архитектуры Laravel и грамотное использование его инструментов позволили обеспечить надежную и эффективную работу приложения.

В результате данной работы, я приобрел ценные навыки и знания в области веб-разработки, что позволит мне успешно реализовывать будущие проекты. Система управления персоналом и рабочими процессами является примером того, как современные технологии могут эффективно решать задачи управления и оптимизации бизнес-процессов.

3.СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Демеле В. [Электронный ресурс]"Laravel: Полное руководство" — URL: [https://www.piter.com/product/laravel-polnoe-rukovodstvo-2-e-izdanie]

(дата обращения:12.05.2024).

1. РозентальН. [Электронный ресурс]"Learning Bootstrap 4" — URL: [<https://for-web.ru/posobie-bootstrap/>] (дата обращения: 19.05.2024).
2. Форсталл М. [Электронный ресурс]" Laravel и AngularJS. Разработка интерактивных веб-приложений” — URL:[ https://bhv.ru/product/laravel-bystraya-razrabotka-sovremennyh-dinamicheskih-web-sajtov-na-php-mysql-html-i-css/] (дата обращения 02.06.2024 )