

## Техническое задание

### Клонировать интернет ресурс в сайт на Laravel

Цель работы: клонировать ресурс Mitre АТТ@СК Enterprise v7.2 в сайт на Laravel

Задачи:

1. Создать схему базы данных
2. Автоматизировать процесс импорта данных в базу, включая обновления, средствами laravel.
3. Создать комплект страниц (blade) аналогичных оригинальному сайту

Для выполнения работ исполнителю предоставляется доступ к базовому приложению на базе которого необходимо реализовать задачу.

#### Базовое приложение

Технические характеристики

- Фреймворк Laravel
- Верстка bootstrap
- Используемые JS компоненты на фронтенде jquery, bootstrap, datatables, select2, daterangepicker, moment.js, pnotify
- Используемые пакеты composer: laravel-datatables, livewire

Приложение развернуто на VDS, опубликовано в интернет.

Будут предоставлены доступ по ssh и к phpmyadmin.

Приложение хранится в приватном репозитории github, настроена связь с экземпляром на VDS.

Будет предоставлен доступ к репозиторию для учетной записи исполнителя на github (для заливки изменений и результатов работ).

Работу можно вести как на предоставленном VDS так и разворачивать приложение у себя, как удобнее.

Установка и использование для работы дополнительных компонентов – только по согласованию.

#### Логика базового приложения

Не все элементы базового приложения нужны для выполнения задачи.

Есть компании (организации). В компаниях пользователи.

Пользователи авторизуются и им становятся доступны объекты компании.

Объекты есть 3 видов: статьи (posts), продукты (products), меры (Actions). Связь между posts и Products многие ко многим.

Объекты могут быть в 2 состояниях - приватные (доступны только конкретной компании) и публичные (доступны всем компаниям и неавторизованным пользователям).

Пользователи могут создавать/изменять/удалять объекты своей компании. Пользователи могут публиковать объекты своей компании или добавлять в свою компанию публичные объекты, при этих операциях создаются новые экземпляры объектов и между ними устанавливается связь.

Список всех компаний и пользователей на главной странице, пароль у всех password.

#### Приемка работ

По результатам работ необходимо:

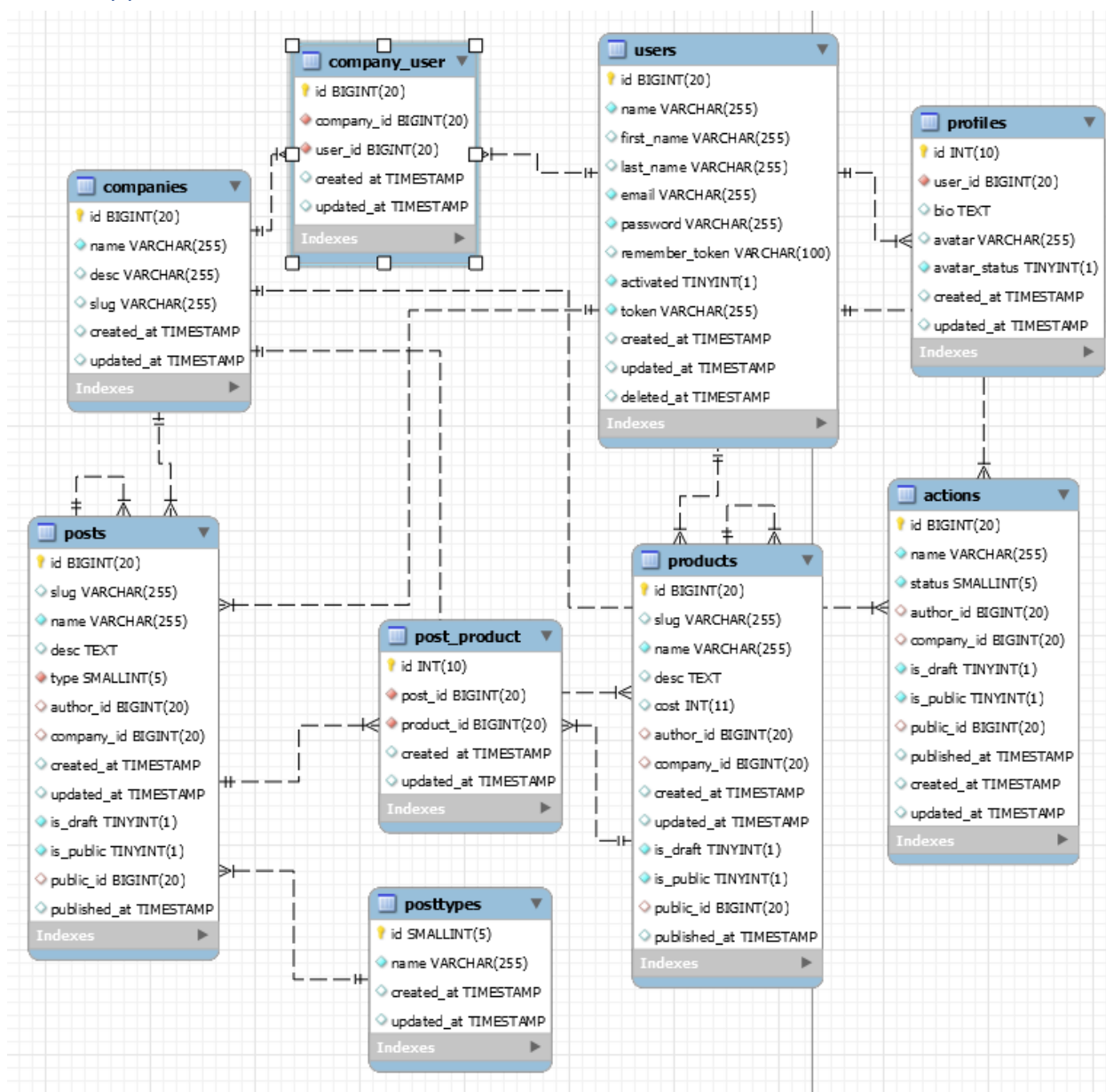
- Продемонстрировать работоспособность (выполнение ТЗ) на предоставленном VDS сервере заказчика.
- Залить все изменения на Github. Не перезаписать репозиторий, а закоммитить, чтобы было видно, какие именно изменения были внесены в код.

В рамках приемки работ будут протестированы механизмы обновления базы. Для чего будет загружена база Mitre ATT@СК Enterprise v7.2.1 содержащая изменения: в описаниях объектов, в связях объектов между собой.

## Оплата

Как удобнее исполнителю. При выполнении первого задания предоплата не предоставляется. Возможен аванс в середине работ при демонстрации части результатов.

## Схема БД



(служебные таблицы не приведены)

## Общие требования к чистоте кода:

- Не использовать сторонние пакеты за исключением обозначенных в ТЗ или дополнительно согласованных.
- Минимизировать количество обращений к базе (проблема n+1 и т.п.).
- Добавлять комментарии на сложных участках кода (лучше на английском, но не обязательно).

- PSR приветствуется.

## Задание

### 1. Создать схему базы данных

Парсить сайт не нужно, все данные есть в json

<https://github.com/mitre/cti/tree/master/enterprise-attack>

<https://github.com/mitre/cti/releases/tag/ATT%26CK-v7.2>

Нужно проработать схему базы данных, чтобы она повторяла оригинальную.

Если в JSON какой то дочерний элемент объекта в виде списка, то он и в базе должны быть в виде отдельной таблицы с связью многие ко многим. Размещать внутри поля списки объектов неправильно.

Названия таблиц с префиксом *mitre\_*

На выходе - комплект файлов миграций laravel

### 2. Автоматизировать процесс импорта данных в базу, включая обновления

Данные в базу должны загружаться через логику приложения, контроллер. Страница, на которой подгружаешь файлы (архив) и запускаешь импорт.

Если данные в базе уже есть - осуществляется обновление данных.

Это нужно для развертывания в будущем новых версий базы ресурса (v.7.3 и т.д.) или развертывания базы на новых экземплярах приложения. Конечно, при условии неизменности структуры данных.

Контроллер, отвечающий за импорт данных должен:

1. Проверять наличие элемента в базе
2. Если элемент уже есть – обновлять его, если нет - создавать.
3. Обновлять (добавлять/удалять) связи элементов, если они изменились.

Пример 1: для элементов таблицы enterprise-attack.json индексом будет поле "external\_id". При повторном заливе базы нужно находить элемент по его external\_id и обновлять его поля (например, description или ссылки на источники)

Пример 2: таблица enterprise-attack.json потребует создания нескольких других таблиц. Например таблицы x\_mitre\_data\_sources. В таких таблицах будут только значения, они уникальные. Эти значения и будут искаться при повторном заливе базы в качестве индексов.

Важно не стирать и не создавать базу заново, а обновлять ее элементы, т.к. к ним по id будут привязаны другие модули сайта.

### 3. Верстка и блейды Laravel

Маршруты как на оригинальном сайте. Но все в разделе site.name/mitre-attack/

Верстку сделать как на оригинальном сайте. Дизайн досконально повторять не обязательно.

Должны получиться следующие страницы:

- Матрица <https://attack.mitre.org/matrices/enterprise/#>

- Достаточно только flat layout. Sub-technics должны отображаться и скрываться интерактивно по нажатию.
- Сделать фильтрацию по Platforms, как на сайте. Получится site.name/mitre-attack/matrices/enterprise/cloud/gcp/
  - Страница с списком тактик <https://attack.mitre.org/tactics/enterprise/>
  - Страницы тактик <https://attack.mitre.org/tactics/TA0001/>
  - Страница техник <https://attack.mitre.org/techniques/enterprise/>
  - Страницы техник <https://attack.mitre.org/techniques/T1548/>
  - Страницы суб техник <https://attack.mitre.org/techniques/T1548/001/>
  - Страница с списком mitigations <https://attack.mitre.org/mitigations/enterprise/>
  - Страницы mitigations <https://attack.mitre.org/mitigations/M1036/>
  - Страница с списком групп <https://attack.mitre.org/groups/>
  - Страницы групп <https://attack.mitre.org/groups/G0053/>
  - Страница с списком софта <https://attack.mitre.org/software/>
  - Страницы софта <https://attack.mitre.org/software/S0069/>

Не забыть про меню, реализовать его в левом меню базового приложения.

Должна быть сделана страница для загрузки/обновления базы. На ней, скорее всего, форма для загрузки zip архива с json файлами. Как альтернативный вариант – архив/каталог с новым комплектом json файлов, размещается на сервере в storage а на странице загрузки указывается каталог и запускается командой загрузка/обновление.

Делайте архитектуру с расчетом на добавление в нее в рамках следующих заданий других матриц - PRE-ATT&CK и Mobile