

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«Інформаційна система підтримки волонтерських  
проектів СумДУ»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Шовкопляс О. А.**

**Студента групи ІН – 63**

**Бублій О.О.**

**СУМИ 2020**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедри Довбиш А.С.

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
до випускної роботи**

Студентки четвертого курсу, групи ІН-63 спеціальності «Інформатика»  
денної форми навчання Бублій Олександрі Олександрівни.

**Тема: «Інформаційна система підтримки волонтерських проєктів  
СумДУ»**

Затверджена наказом СумДУ

№ \_\_\_\_\_ від \_\_\_\_\_ 20\_\_ р.

**Зміст пояснювальної записки:** 1) огляд існуючих рішень; 2) аналіз проблеми; 3) постановка задачі; 4) проєктування системи; 5) програмна реалізація; 6) аналіз роботи.

Дата видачі завдання « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Керівник випускної роботи \_\_\_\_\_ Шовкопляс О. А.

Завдання приняла до виконання \_\_\_\_\_ Бублій О. О.

## РЕФЕРАТ

**Записка:** 55 стор., 14 рис., 3 додатки, 12 джерел.

**Об'єкт дослідження** — організація робочого процесу та розповсюдження важливої інформації за допомогою інформаційних систем.

**Мета роботи** — аналіз та створення ефективної інформаційної системи підтримки волонтерських проєктів СумДУ в котрій буде сконцентровано та впорядковано всю необхідну для оприлюднення інформацію.

**Методи дослідження** — об'єктно-орієнтоване проєктування.

**Результати** — розроблено інформаційну систему для підтримки волонтерських проєктів СумДУ з використанням архітектурного шаблону програмного забезпечення клієнт-сервер, за принципами даної архітектури було створено функціонуючі компоненти нашої системи, а саме: серверна частина системи, WEB сайт для взаємодії з сервером за допомогою інтернет браузера. Розроблені компоненти реалізовано з використанням мови програмування PHP та програмного середовища Laravel.

ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ВОЛОНТЕРСЬКИХ  
ПРОЄКТІВ СУМДУ, КОНЦЕНТРАЦІЯ ТА РОЗПОВСЮДЖЕННЯ  
ІНФОРМАЦІЇ ПРО ВОЛОНТЕРСТВО В СУМДУ, АРХІТЕКТУРНИЙ  
ШАБЛОН КЛІЄНТ-СЕРВЕР, MODEL VIEW CONTROLLER.

## ЗМІСТ

ВСТУП.....	5
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ. АНАЛІЗ ПРОБЛЕМИ. ПОСТАНОВКА ЗАДАЧІ.....	6
1.1 Розробка вимог до ІС .....	6
1.2 Аналіз архітектури та технологій створення існуючих систем .....	7
1.4 Огляд літератури на тему створення подібних систем .....	8
2 ПРОЄКТУВАННЯ СИСТЕМИ .....	10
2.1 Визначення архітектури та технологій реалізації .....	10
2.2 Робота з API і WEB маршрутами .....	11
2.3 Проектування інтерфейсу Web додатка.....	13
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ РОБОТИ.....	15
3.1 Структура файлів .....	15
3.3 Створення бази даних та Web сторінок .....	18
3.4 Перевірка роботи Web додатка.....	20
ВИСНОВКИ .....	24
СПИСОК ЛІТЕРАТУРИ.....	26
ДОДАТОК А Програмний код проєкту створення API.....	27
ДОДАТОК Б Програмний код проєкту бази даних та міграцій .....	48
ДОДАТОК В Папка socstor .....	57

## ВСТУП

Сьогоднішній світ містить в собі безмежні потоки інформації різних видів. При цьому її кількість у будь якій сфері нашого життя постійно збільшується. А відтак і волонтерський рух у Сумському державному університеті, котрий має тривалу історію, з кожним роком має все більше бажаючих вступити до його лав. Необхідно розуміти, що важливим є не тільки факт донесення інформації до студентів, а і її здатність характеризувати поточну ситуацію, форма подачі, здатність забезпечити суб'єкта всіма необхідними даними за найкоротший проміжок часу.

Для вирішення цієї проблеми викладачі, студенти та просто зацікавлені в добродійності люди вимушені проводити пошук потрібної інформації в газетних статтях, оголошеннях не просторах інтернету та різноманітних соціальних мереж. Це дуже не зручно, а головне часозатратно, хоча всю цю інформацію об'єднує наша прагматика. Кожен окремий ресурс може добре вирішити одну з проблем (здатність характеризувати поточну ситуацію, зручність форми подання, доступ до інформації у будь який проміжок часу), але не всі разом.

Таким чином, через актуальність наведеної проблеми метою роботи є створення ефективної інформаційної системи підтримки волонтерських проєктів СумДУ, яка дозволить всім бажаючим, зацікавленим у добродійності, своєчасно отримувати актуальну інформацію щодо соціальної роботи СумДУ.

# 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ. АНАЛІЗ ПРОБЛЕМИ. ПОСТАНОВКА ЗАДАЧІ

## 1.1 Розробка вимог до ІС

Для ефективного вирішення актуальних проблем підтримки волонтерських проєктів, необхідне визначення якості кожної складової, використовуючи множину показників, що характеризують стан ІС [1].

Не менш важливим постає питання формування задач роботи, котрі необхідно виконати задля досягнення поставленої мети, а саме:

1. Детальне дослідження існуючих рішень, методів роботи та технологій їх створення.
2. Проведення ознайомлення з необхідною літературою та аналіз даних з проєктування подібних систем та технологій їх створення.
3. Вибір технологій для подальшого проєктування та створення інформаційної системи.
4. Проєктування системи комунікаційного координаційного центру.
5. Реалізація системи за допомогою використання визначених технологій.

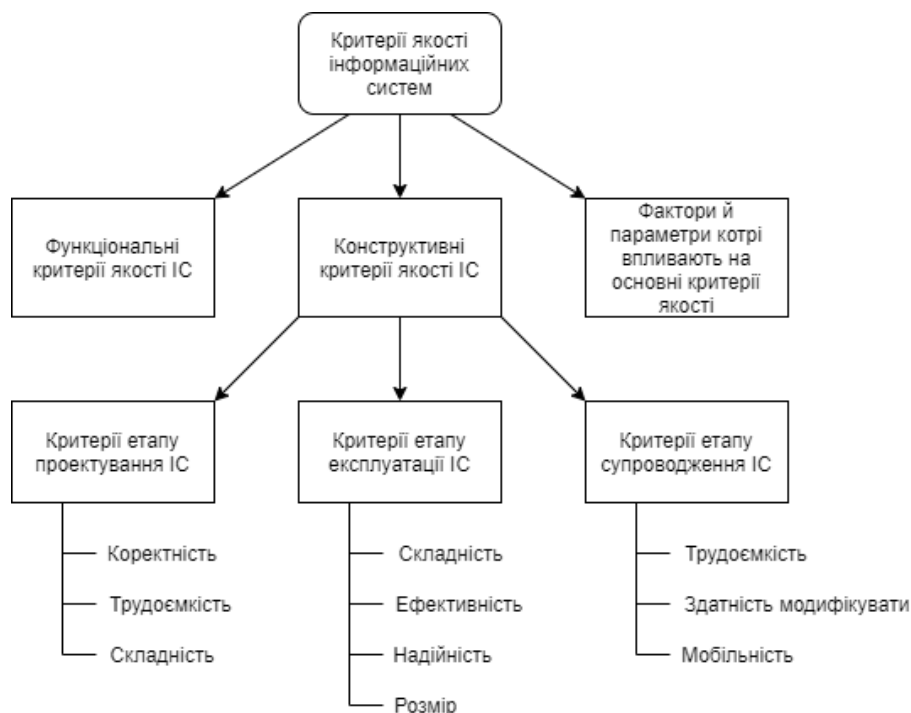


Рисунок 1.1 - Класифікація критеріїв якості ІС

Таким чином, оцінюючи якість складових ІС, можна зробити висновок про якість функціонування ІС в цілому і отримати можливість управління її якістю (рис. 1.2).

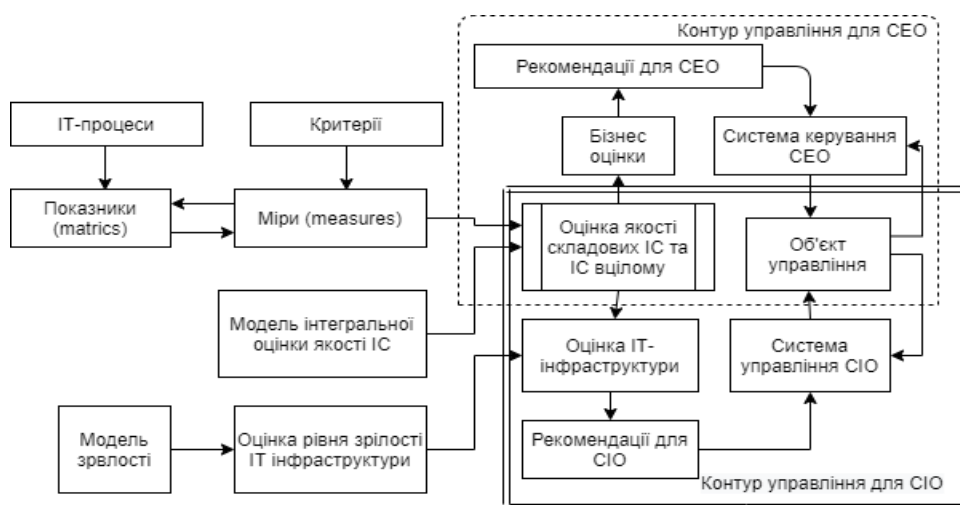


Рисунок 1.2 - Модель управління якістю ІС

Ядром даної моделі управління якістю ІС є процес оцінки якості складових ІС.

## 1.2 Аналіз архітектури та технологій створення існуючих систем

Найбільшою популярністю серед студентів та викладачів користуються інформаційні системи зі зручним та зрозумілим поданням інформації. Прикладом подібних систем можуть послужити певні сайти та веб-сторінки в просторах інтернету. Їх перевага полягає в простоті використання, швидкодії та можливості користуватися ними за допомогою різних гаджетів.

Всі вони мають подібні принципи роботи та структуру подачі інформації користувачам. Такий метод організації зручний для розміщення, з метою оприлюднення, будь якої інформації об'єднаної за змістом та спрямованістю. Таким чином перед розробниками стоїть задача коректно організувати роботу сайта та зручність його використання для користувачів.

Переважає більшість таких інформаційних систем використовують підхід з клієнтно-серверною архітектурою в якій архітектурні шаблони виражають фундаментальну схему структурної організації певної програмної

системи. Така схема складається із визначених наперед підсистем, а також точно визначає їхні сфери відповідальності та взаємовідносини. А це перш за все ефективне управління вмістом, чітко визначені процедури та інформаційна політика сайту. Всі елементи цих систем проходять регулярні тестування.

Серверні частини більшості таких систем побудовані за допомогою фреймворків написаних на мові програмування PHP - скриптова мова програмування. Це дозволяє їм працювати з будь-якими операційними системами. Вона є однією з найпоширеніших мов, що використовуються у сфері веб-розробок.

#### **1.4 Огляд літератури на тему створення подібних систем**

Задля вирішення проблеми, перед нами стоїть конкретне завдання об'єднати всю існуючу інформацію. За час свого існування модель клієнт-серверної архітектури не зазнала суттєвих змін та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Процес її розвитку був і залишається еволюційним.

За час такої еволюції з'явилися багато нових рішень що дозволили використовувати модель клієнт-сервер для рішення нових задач. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Наступним важливим питанням постає вибір долучної моделі тестування. Вона має глобальний вплив на процес створення інформаційних систем [2, 5]. Точки зору розділяються стосовно пріоритетності тестування в процесі розробки. В залежності від складності інформаційної системи, її архітектури, методології розробки, кількості доступних ресурсів, способи



варіюються від використання керованої тестами розробки (test driven development) до ручного тестування.

Відомо, що чим раніше помилку виявлено, тим легше її локалізувати і виправити. Автотести у багатьох випадках допомагають скоротити час виявлення помилки до мінімуму, завдяки чому помилка виправляється за гарячими слідами.

Також необхідне дослідження патернів як інструменту проєктування інформаційних систем, ааналіз методів проєктування, пошук проблеми проєктування і експлуатації інформаційної систем. Адже правильний вибір паттерну напряду зв'язаний з якістю нашого продукту. Всі вони мають свої переваги та недоліки. Застосування патернів має передувати вивчення інформаційної ситуації [6], в якій знаходиться проєктована інформаційна система.

Концепція MVC (Model-View-Controller: модель-вид-контролер) дуже часто згадується в світі веб програмування. Завдяки своїй зрозумілості та простоті реалізації паттерн MVC слугує стартовою точкою для створення багатьох додатків [7]. Шалена популярність даної структури в веб додатка склалася завдяки її включенню в два середовища розробки, які стали дуже популярними: Struts і Ruby on Rails. Ці дві середовища розробки намітили шляхи розвитку для сотень робочих середовищ, створених пізніше.

## 2 ПРОЄКТУВАННЯ СИСТЕМИ

### 2.1 Визначення архітектури та технологій реалізації

Для створення головних моделей використовується фреймворк Laravel. Taylor Otwell (творець Laravel), взяв найкраще з існуючих PHP фреймфорків, а також Ruby on Rails, ASP.NET MVC, Sinatra і створив фреймворк, котрий максимально просто вирішує повсякденні завдання програмістів.

Laravel - це full stack рішення, як для back-end так і для front-end розробників. Для фронтенда є система коробки -збирання laravel Mix, яка побудована на Webpack, а також js фреймворк - Vue.

Наявність інтегрованої системи модульного тестування phpunit, що робить наш сайт більш надійним.

Безпека. Можливість отримати несанкціонований доступ до бази даних вкрай складно. Високий рівень безпеки гарантує надійний захист від SQL-injection, атак типу XSS, CSRF.

Починаючи з 2013 року Laravel визнаний кращим і найпопулярнішим фреймфорком в світі. Результати порівнянь на рисунку 2.1.



Рисунок 2.1 – Популярність фреймворку у персональних проєктах

Для розробки Frontend буде використано фреймворк vue.js. Vue.js - JavaScript-фреймворк з відкритим вихідним кодом для створення користувацьких інтерфейсів. Легко інтегрується в проекти з використанням інших JavaScript-бібліотек. Може функціонувати як веб-фреймворк для розробки односторінкових додатків в реактивному стилі. Vue.JS досить успішно конкурує з React.JS - очевидним лідером в області View. Із сучасних популярних технологій, вирішальних схожі завдання можна також виділити Angular і Ember.

Створення якісних онлайн-ресурсів різного масштабу або ж односторінкових сайтів-лендінгів обов'язково вимагає грамотної верстки та реалізації професійного дизайну. Думки веб дизайнерів про сітку Bootstrap сходяться - це відмінний засіб для вирішення проблем, пов'язаних з розширенням бізнесу і отриманням на виході чудового результату. Практична і функціональна сітка. Bootstrap являється унікальним вільним набором найважливіших інструментів для реалізації створення сайтів і унікальних веб-додатків [8]. Цей додаток обов'язково включає в себе:

- ✓ гарантовано якісні HTML і CSS-шаблони оформлення для грамотної типографіки;
- ✓ створює веб-форми;
- ✓ кнопки і мітки;
- ✓ блоки визначення навігації та інші компоненти функціонального веб-інтерфейсу;
- ✓ встановлює JavaScript-розширення.

## **2.2 Робота з API і WEB маршрутами**

Усі маршрути Laravel визначені у rout-файлах та містять сотні шляхів, окремимих секцій для гостей, користувачів, адміністраторів і т.д. Знаходяться вони в каталозі маршрутів. Ці файли автоматично

завантажуються фреймворком. Їх групування та додання префікси в URL відбувається в Laravel за замовчуванням, за допомогою двох файлів:

- routes/web.php;
- routes/api.php.

Файл routes/web.php визначає маршрути, призначені для нашого веб-інтерфейсу. Ці маршрути призначаються груповим посередницьким програмним забезпеченням, яке забезпечує такі функції, як стан сесії та захист CSRF, тобто вони організовані та структуруються по групах. Основне використання - це приховування різних груп під різними мідлварами. При цьому використовуються імена і префікси груп маршрутів (рис. 2.2).

```
76 Route::prefix('admin')->group( function () {
77     Route::get('/login', 'Auth\AdminLoginController@showLoginForm')->name('admin.login');
78     Route::get('/logout', 'Auth\AdminLoginController@logout')->name('admin.logout');
79     Route::post('/login', 'Auth\AdminLoginController@login')->name('admin.login.submit');
80     Route::resource('/configs', 'Admin\ConfigController', ['as' => 'admin'])->except([
81         'create', 'show', 'store', 'destroy'
82     ]);
83     Route::get('/', 'Admin\AdminController@index')->name('admin.dashboard');
84     Route::group(['prefix' => 'user-management', 'namespace' => 'Admin\UserManagement'], function() {
85         Route::resource('/users', 'UserController', ['as' => 'admin.user-management'])->except('show');
86         Route::resource('/admins', 'AdminController', ['as' => 'admin.user-management'])->except('show');
87     });
88 });
```

Рисунок 2.2 - Структуризація по групах

У файлі routes/api.php реєструються маршрути API для нашої програми та є безстанними. Ці маршрути завантажуються RouteServiceProvider в межах групи, котра присвоюється групі проміжних програм "api". У цій групі префікс /api URI автоматично застосовується, тому не потрібно вручну застосовувати його до кожного маршруту у файлі [9]. Код основних файлів проєкту розміщено в додатку А.

## 2.3 Проектування інтерфейсу Web додатка

Прототипи інтерфейсу Web додатка зображено на рисунках 2.3 та 2.4.



Рисунок 2.3 - Прототип сторінок мобільної версії Web додатка

Створення логіки і стилю веб-продукта є основними завданнями веб-дизайнера. Вони мають зробити сайт зручним та приємним для користування. Є список умовних правил на яких слід зацентувати увагу під час роботи над веб-проектом. Вони будуть корисними не лише для дизайнерів за професією, а й для тих, хто намагається самостійно задовільнити декілька ролей при створенні сайту.

По перше, інтерфейс має бути однорідним для всього проєкту. Однорідність та узгодженість інтерфейсу на всіх сторінках – це головний принцип добре продуманого проєкту. Схеми, шрифти, стиль правопису, позитивно впливають на зручність користування і на користувацький досвід.

Також сторінка має бути легкою для сканування очима. Зпершу користувач, швидше за все, бігло огляне вміст сторінки, аніж прочитає її повністю. Більш того, подібне швидке сканування буде відбуватися, допоки користувач не знаєде бажану інформацію. Ця проблема може бути вирішена за допомогою візуальної ієрархії. Тобто позначаються місця, де потрібно акцентувати увагу в першу чергу, в другу і так далі.



Рисунок 2.4 - Прототип сторінок комп'ютерної версії Web додатка

Хоча мобільна версія сайту часто виглядає більш просто, але підхід до процесу її створення є не менш відповідальним, як і до розробки основного сайту. Адже з сучасним ритмом життя саме наші смартфони постійно знаходяться в наших руках, де б ми не були: вдома, на роботі, навчанні або відпочинку.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ РОБОТИ

### 3.1 Структура файлів

Кореневий каталог:

- App. Папка app містить код ядра застосування, майже всі класи програми будуть знаходитися в цій папці.
- Bootstrap. Папка bootstrap містить файли, які завантажують фреймворк і налаштовують автозагрузку. Також в папці bootstrap знаходиться папка cache, яка містить згенеровані фреймворком файли для оптимізації продуктивності - наприклад, кеш-файли маршрутів і сервісів.
- Config. Папка config містить всі конфігураційні файли додатків.
- Database. Папка database містить міграції і класи для наповнення початковими даними БД. При необхідності цю папку можна використовувати для зберігання бази даних SQLite.
- Public. Папка public містить файл index.php, який є вхідний точкою для всіх запитів, що надходять в програму. Також ця папка містить ресурси, такі як зображення, JavaScript, CSS.
- Resources. Папка resources містить уявлення, а також сирі, некомпільовані ресурси, такі як LESS, SASS, JavaScript. А також тут знаходяться всі «мовні» файли.
- Tests. Папка tests містить Автотести. Спочатку там вже є приклад PHPUnit. Клас кожного тесту повинен мати в імені суфікс Test. Можна запускати свої тести командами `phpunit` і `php vendor / bin / phpunit`.
- Storage. Папка storage містить скомпільовані Blade-шаблони, файл-сесії, кеші файлів і інші файли, створювані фреймворком. Ця папка ділиться на підпапки `app`, `framework` і `logs`. В папці `app` можна зберігати будь-які файли, які генеруються додатком. В папці `framework` зберігаються створювані фреймворком файли і

кеш. А в папці `logs` знаходяться файли журналів програми. Папку `storage / app / public` можна використовувати для зберігання призначених для користувача файлів, таких як аватарки, які повинні бути доступні всім. Створюється символічне посилання на `public / storage`, яка веде до цієї папки. Ви можете створити посилання командою `php artisan storage:link`.

Каталог `app`. Основна частина програми знаходиться в каталозі `app`. За замовчуванням цей каталог зареєстровано за простором імен `App` та автоматично завантажено за допомогою `Composer` за стандартом автозавантаження `PSR-4`.

В каталозі `app` знаходиться ряд додаткових каталогів, таких як `Console`, `Http` та `Providers`. Можна сказати, що каталоги `Console` і `Http` надають АРІ ядра вашого застосування. Протокол `HTTP` і командний рядок - це механізми взаємодії з вашим додатком, але вони не містять логіки додатка. Іншими словами, це просто два способи передачі команд вашому додатку. Каталог `Console` містить всі ваші `Artisan`-команди, а каталог `Http` містить ваші контролери, посередники і запити.

Багато інших каталогів будуть створені в каталозі `app`, коли ви виконаєте `Artisan`-команду `make` для генерування класів. Наприклад, каталог `app / Jobs` не буде створена, поки ви не виконаєте `Artisan`-команду `make:job`, щоб згенерувати клас завдання. :

- `Console`. Папка `Console` містить всі додаткові `Artisan`-команди для застосування. Ці команди можна згенерувати командою `make:command`. Також цей каталог містить ядро вашої консолі, де реєструються ваші додаткові `Artisan`-команди і визначаються ваші заплановані завдання.
- `Exceptions`. Папка `Exceptions` містить обробник виключень вашого застосування. Ця папка також є хорошим місцем для розміщення всіх винятків, що виникають у додатку. Якщо є потреба зміни



того, як журналюються і відображаються ваші виключення, треба змінити клас Handler в цьому каталозі.

- Http. Папка Http містить контролери, посередники і запити форм. Тут буде розміщена майже вся логіка обробки запитів, що входять в додаток.

Providers. Папка Providers містить всі сервіс-провайдери для застосування. Сервіс-провайдери завантажують додаток, прив'язуючи сервіси в сервіс-контейнер, реєструючи події, і виконуючи будь-які інші завдання для підготовки програми до вхідних запитів. У свіжовстановленому додатку Laravel ця папка вже містить кілька провайдерів. При необхідності ви можете додавати свої провайдери в цю папку.

Вся файлова структура проекту зображена на рисунках 3.1 та 3.2 3.1 .

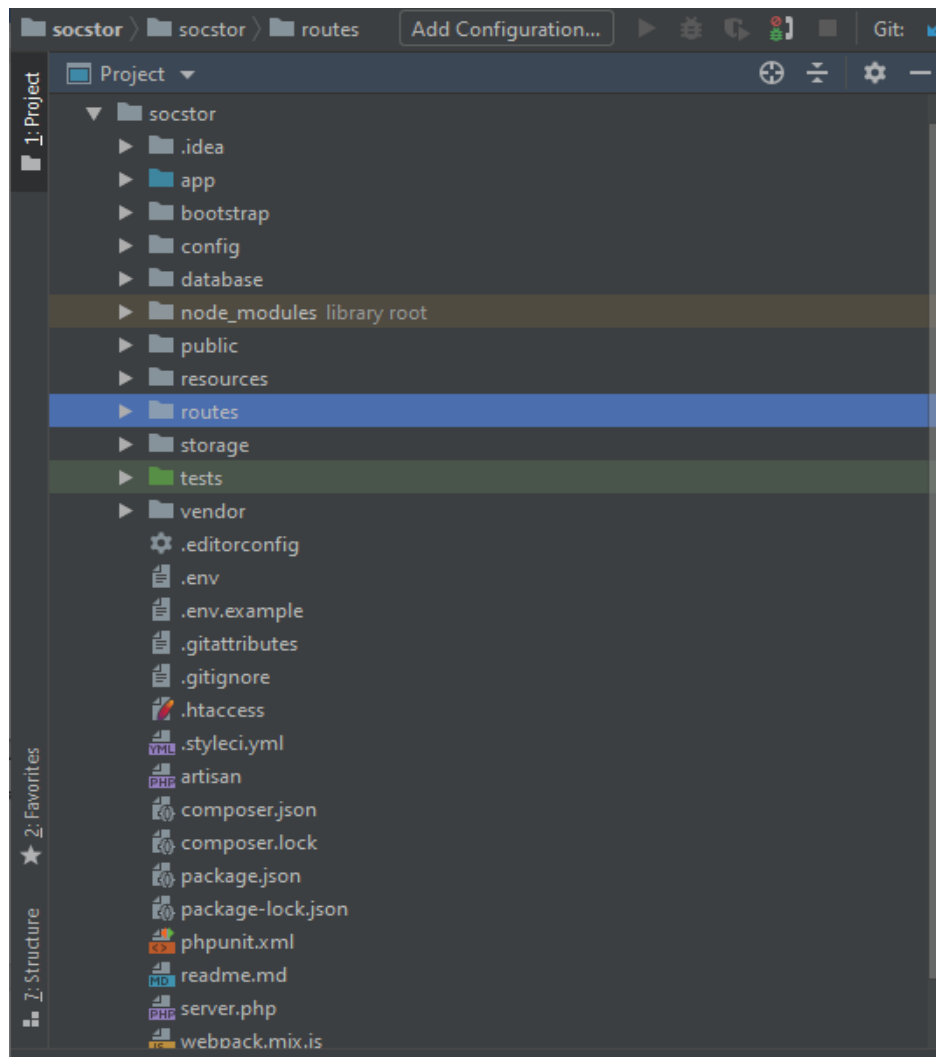


Рисунок 3.1 – Структура файлів проекту

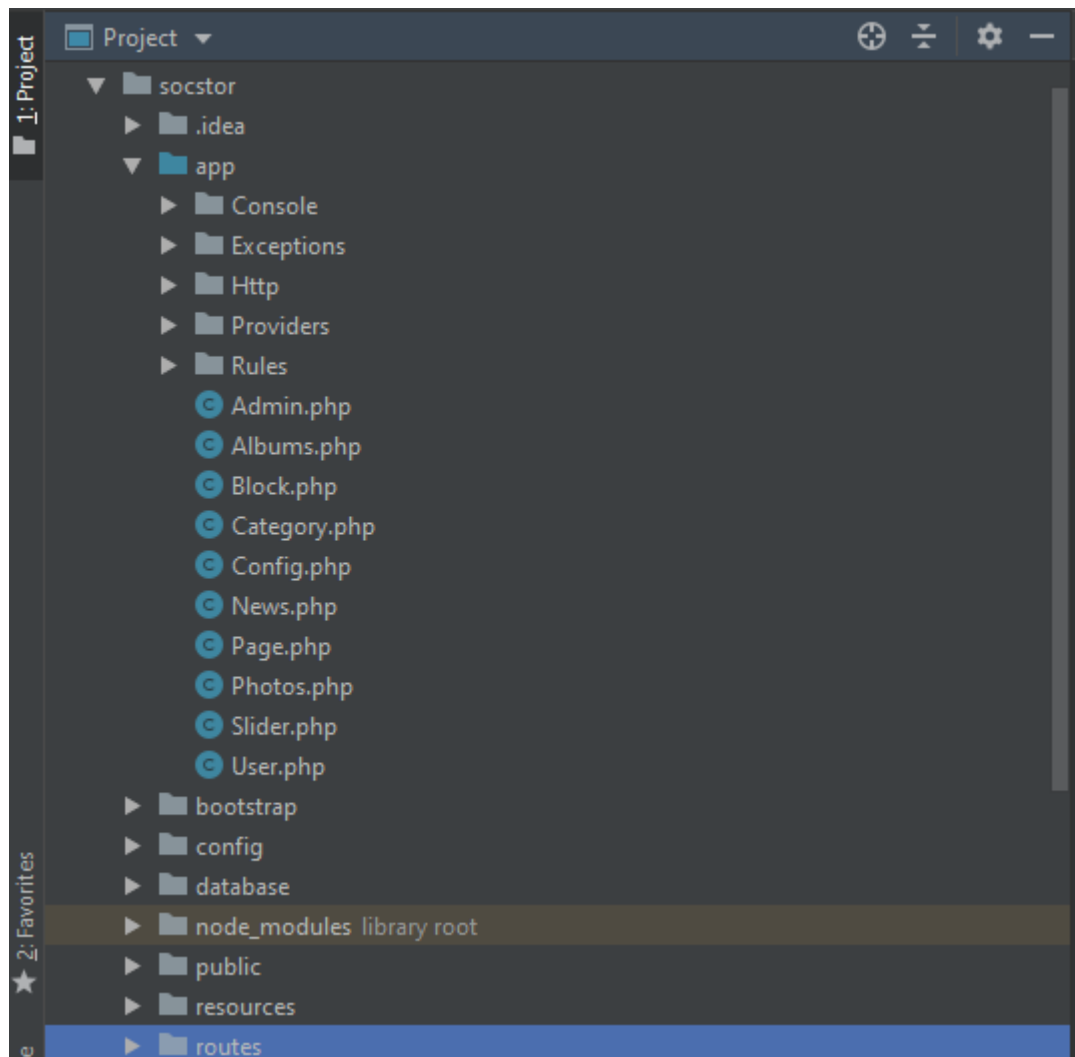


Рисунок 3.2 - Структура файлів папки app

Код основних файлів проєкту створення розміщено в додатку А.

### 3.3 Створення бази даних та Web сторінок

Після створення деяких загальних моделей та основних бази даних, необхідне створення міграцій баз даних та сідів. Традиційно "dump" файли бази даних використовувались як спосіб обходу обох схем, структури таблиць та даних, які були б початковими або заздалегідь визначеними. До них відносяться записи, такі як значення за замовчуванням; незмінні списки, зазвичай міста чи країни; і користувачів, таких як "адміністратор". Цим дамп-файлам, що містять SQL, може бути призначено контроль вихідного коду. Це не завжди найкращий спосіб зберегти цілісність бази даних; оскільки кожен раз, коли розробник додає записи або модифікує базу даних, усім

розробникам в команді необхідно буде скинути і відтворити базу даних, додати або видалити дані, таблиці, рядки, стовпці чи індекси вручну. Міграції дозволяють базам даних жити у вигляді коду, фактично перебуваючи всередині проєкту Laravel [10].

Laravel містить дві міграції в каталозі `database/migrations`. Перша міграція створює таблицю користувачів. Друга - таблицю `password_resets`, яка використовується для відновлення втрачених паролів. Якщо не зазначено, міграції діють в базі даних, яка налаштована у файлі конфігурації `/config/database.php`.

Міграції виконуються з командного рядка і їх також можна автоматизувати для автоматично створення бази даних, коли це потрібно. Якщо вона вже не існує, скидайте та відтворюйте таблиці, або заповнюйте таблиці, якщо вони вже існують.

Міграції розширюють та використовують клас `Blueprint`. Існує два способи: вгору та вниз, які застосовуються під час команд переміщення і команди відкату, використовуються відповідно. Метод `Schema :: create ()` викликається ім'ям таблиці як першого параметра, а функція зворотного виклику як другий параметр, який приймає екземпляр об'єкта `Blueprint`.

Код основних файлів міграцій проєкту розміщено в додатку Б.

### 3.4 Перевірка роботи Web додатка

Перегляд основних сторінок мобільної версії зображений на рисунках 3.3 3.4 та 3.5

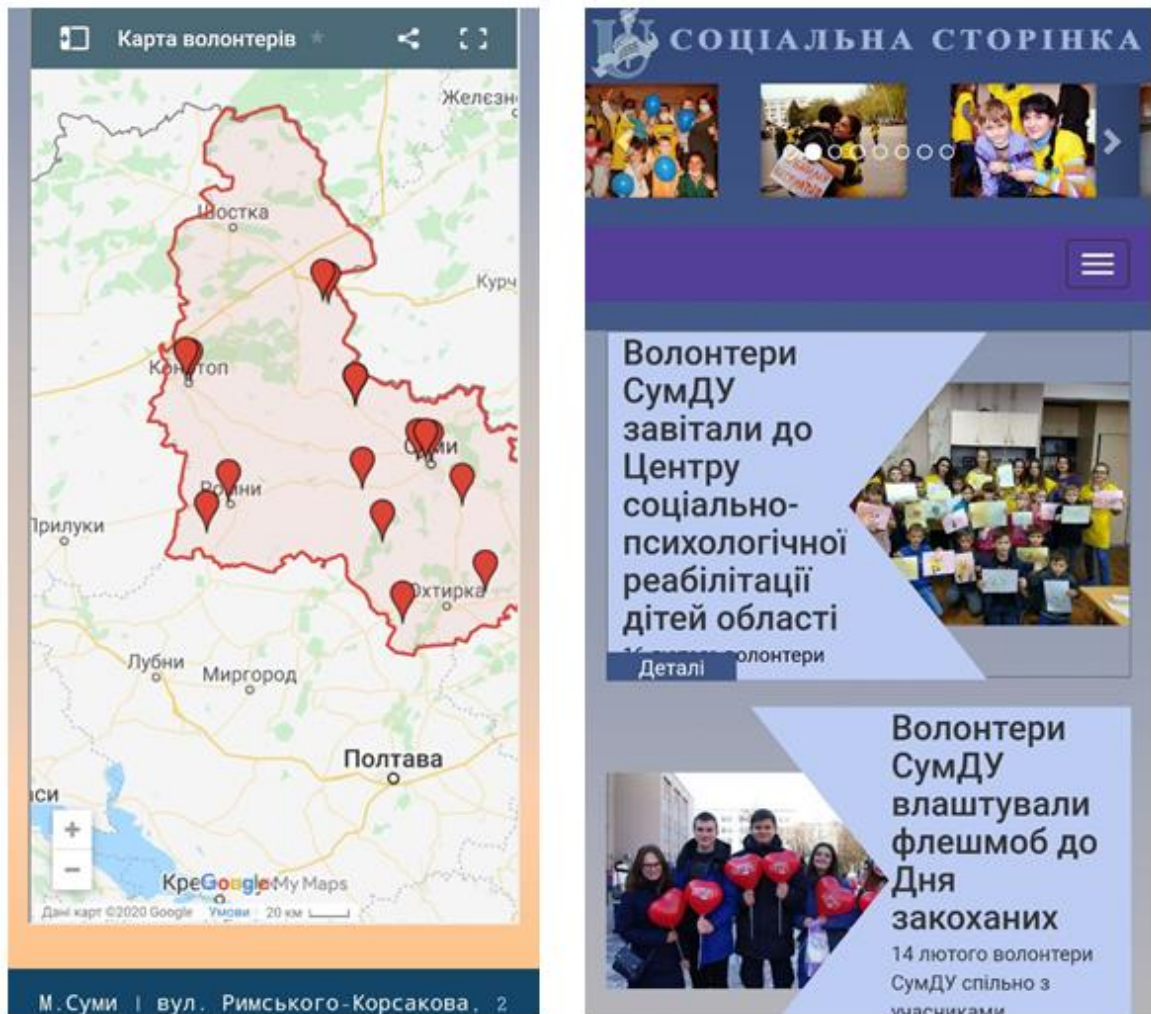


Рисунок 3.3 - Карта волонтерів та голвна сторінка сайту

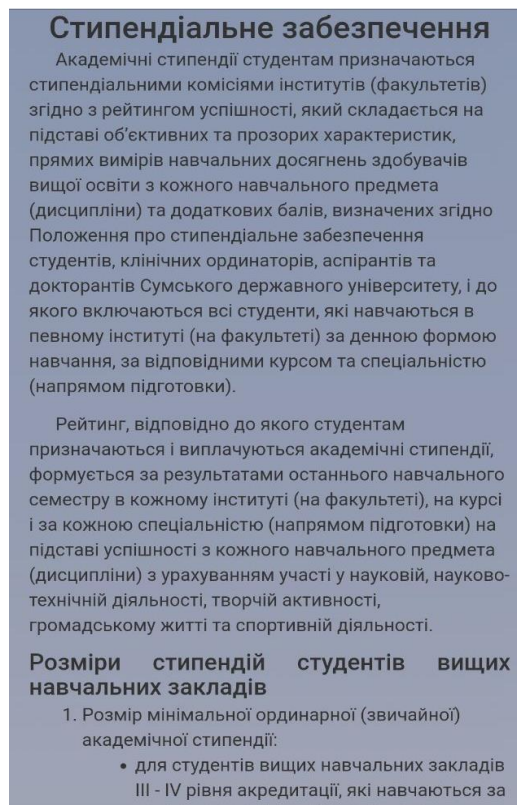


Рисунок 3.4 – Стипендіальне забезпечення (Нормативні документи)

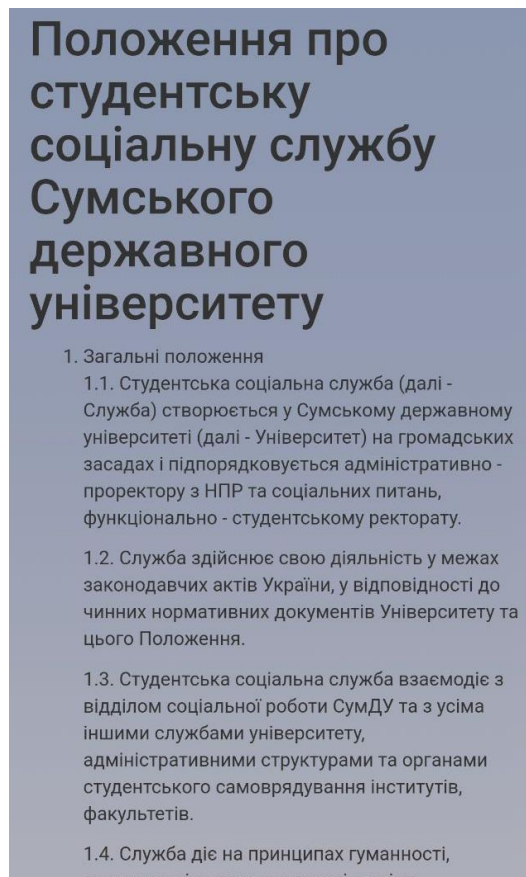


Рисунок 3.5 – Студентська соціальна служба



Перегляд основних сторінок версії для комп'ютера зображено на рисунках 3.6 ,3.7, 3.8 та 3.9 .



Рисунок 3.6 - Головна сторінка

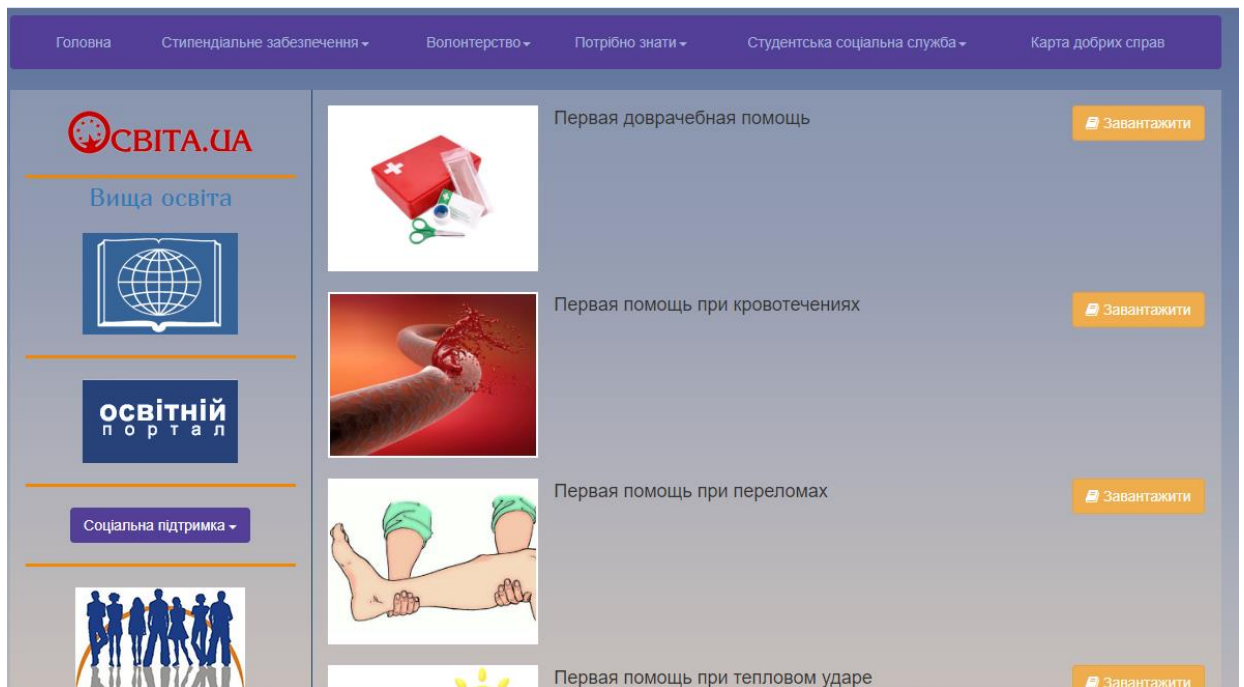


Рисунок 3.7 - Потрібно знати (Корисна інформація)

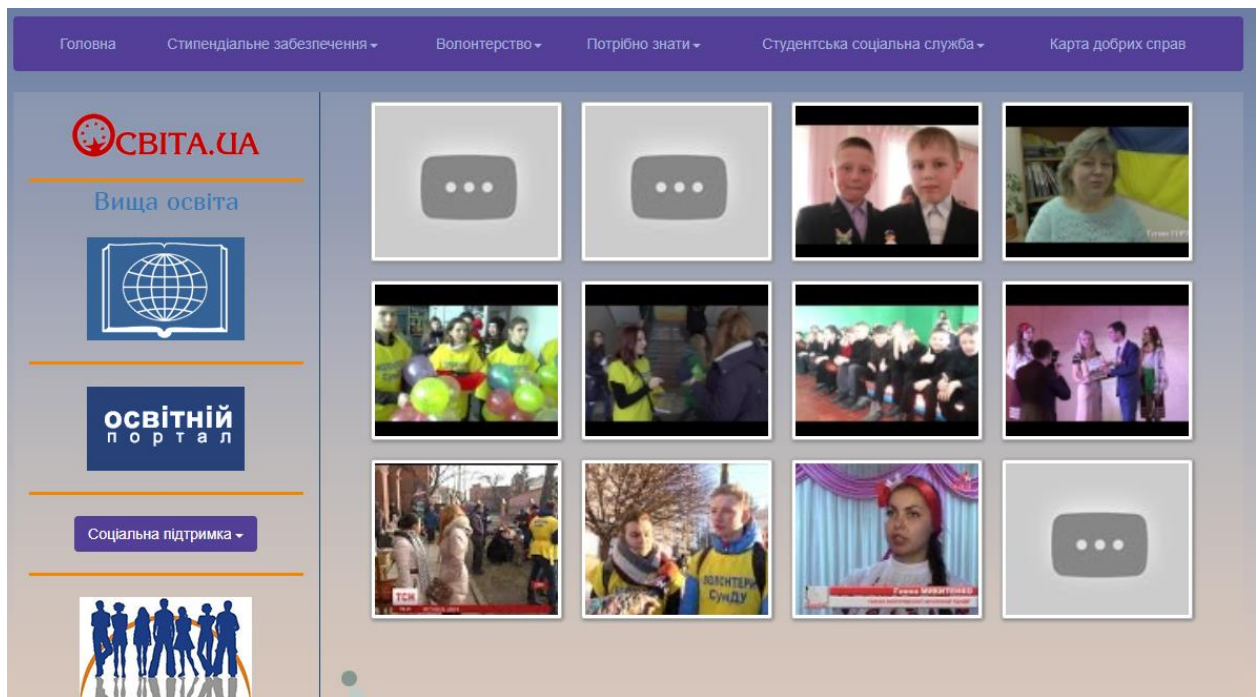


Рисунок 3.8 - Волонтерство (Відео)

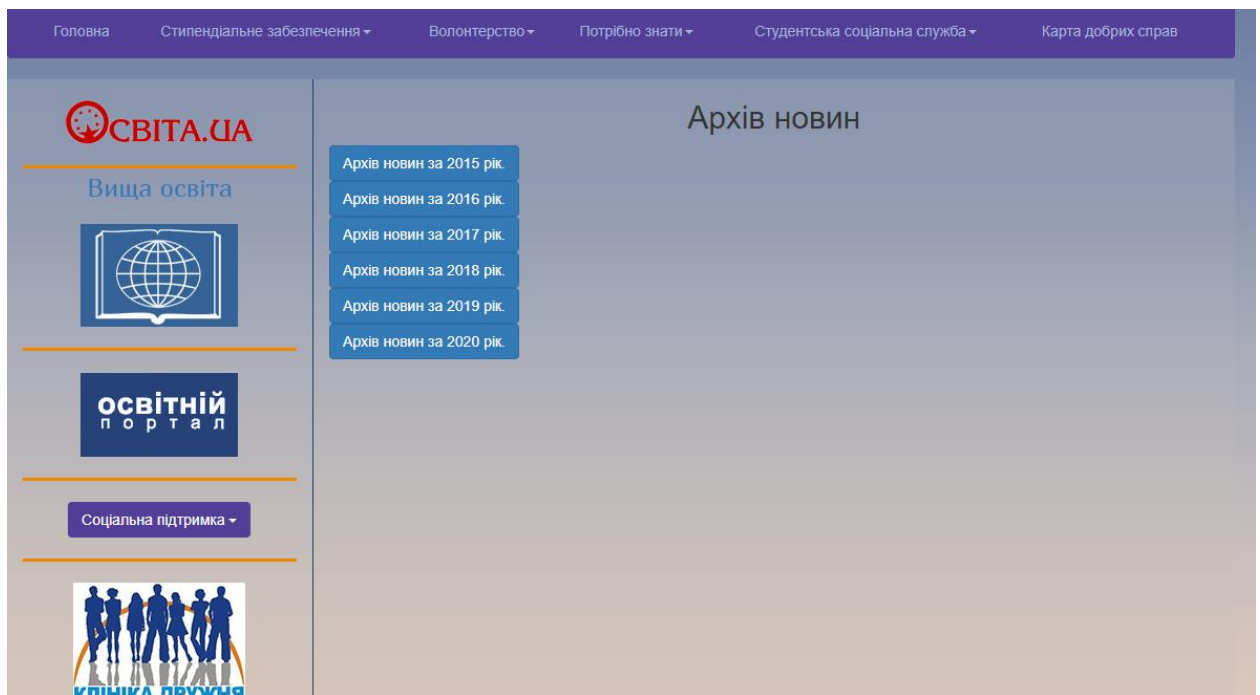


Рисунок 3.9 – Волонтерство (Конкурси)

## ВИСНОВКИ

Протягом дипломного проєктування були виконані всі поставлені задачі, а саме: проведено детальне дослідження існуючих рішень, завдяки якому були встановлені методи роботи подібних систем, їх основні недоліки та переваги, типи архітектури, підтримувані ними операційні системи та технології що використовувались для їх створення. Це дозволило сформулювати загальні вимоги до нашого проєкту, а саме:

1. використання клієнт-серверної архітектури;
2. наявність декількох клієнтських додатків для роботи з серверною частиною системи;
3. використання технологій створення що забезпечують високий рівень швидкодії та надійності системи.

По темі проєктування схожих систем було проведено аналіз літератури, що дало нам можливість коректно визначити переваги використання одних моделей управління, технологій створення, архітектурних шаблонів, методів проєктування і т. д. над іншими.

Відповідно до встановлених умов обмеження часу та ресурсів на розробку інформаційної системи, було визначено перелік методів проєктування та створенню перелік технологій, котрі відрізняються високими показниками безпеки та швидкодії.

Як наслідок, реалізації інформаційної системи з використанням визначених технологій, було отримано:

1. готову до роботи та тестування серверну частину системи;
2. Web додаток (з розширенням як для лептопів так і для мобільних пристроїв );
3. проведено перевірку роботи різних компонентів інформаційної системи, виявлено та виправлено існуючі помилки.

В результаті виконання перелічених робіт отримано надійну та швидко інформаційну систему, яка відповідає визначеним вимогам та стане основою



для проведення робіт з подальшого вдосконалення. В перспективах розвитку планується провести додаткові дослідження, збільшити функціонал системи, покращити рівень безпеки та швидкодію, вдосконалити дизайн та розробити мобільний додаток для операційної системи IOS

Тези на тему «Інформаційна система підтримки волонтерських проєктів СумДУ» увійшли в матеріали міжнародної науково-технічної конференції студентів та молодих вчених “Інформатика, математика, автоматика :: 2020”, секція «Комп’ютерні науки та Кібербезпека».

## СПИСОК ЛІТЕРАТУРИ

1. Александр И. Пушкарь, Владимир В. Гаркин. Показатели и меры оценки качества информационных систем предприятия. // 2014.
2. Лайза Кріспін. Джанет Грегори. Agile-тестирование. Обучающий курс для всей команды. // 2015. – 528 с.
3. Jim Webber, Savas Parastatidis, Ian Robinson. REST in Practice: Hypermedia and Systems Architecture. // 2010. – 415 p.
4. Leonard Richardson. RESTful Web APIs: Services for a Changing World. // 2013. – 406 p
5. В. Романько. Автотесты – барское дело – <https://habr.com>
6. Владимир Александрович Мордвинов. Методические исследования: Использование паттернов для проектирования информационных систем. // ИТНОУ. 2018.
7. Joshua Greene & Jay Strawn // Design Patterns, 2018. – 359 p.
8. Сильвио Морето. Bootstrap в примерах. / Пер. с англ. Рагимов Р. Н. // Науч. ред. Киселев А. Н. – М.: ДМК Пресс, 2017.
9. Laravel: Up and Running: A Framework for Building Modern PHP Apps // 2016.
10. Christopher John Pecoraro. Mastering Laravel. // 2015. – 201 p
11. Tanmay Deshpande. RESTful API Design — Step By Step Guide – <https://hackernoon.com/>
12. W. Jason Gilmore and Eric L. Barnes / Easy E-Commerce Using Laravel and Strip. // 2015. – 110 p.

## ДОДАТОК А

### Програмний код проєкту створення API

Папка Resources:

Файл volunteer.blade.php

```
@extends("layouts.app")

@section('meta_title', $volonter->first_title)
@section('meta_keywords', $volonter->meta_keywords)
@section('meta_description', $volonter->meta_description)

@section('content')
    <div class="page-title">
        <span>{{ $volonter->first_title }}</span>
    </div>

    <div class="info-page">
        <div class="description">
            {!! $volonter->description !!}
        </div>

        <div class="photos">
            @foreach ($lastAlbums as $albums)
                <div class="photo">
                    <a href="{{route('album', ['id' => $albums->id])}}">
                        
                        <div class="album-description">{{ $albums->title }}</div>
                    </a>
                </div>
            @endforeach
        </div>

        <div class="more-news">
            <a href="{{route('albums')}}"><button class="more-albums-btn">
Перейти до фотоальбомів</button> </a>
        </div>
    </div>
@endsection
```

Файл volunteer.blade.php

```
@extends("layouts.app")

@section('meta_title', $map->first_title)
@section('meta_keywords', $map->meta_keywords)
@section('meta_description', $map->meta_description)

@section('content')
    <div class="page-title">
        <span>{{ $map->first_title }}</span>
    </div>
    <div class="info-page">
        <div class="description">{!! $map->description !!}</div>
    </div>
```

```

        <div class="map">
            <iframe
src="https://www.google.com/maps/d/embed?mid=zUazofjOhDWs.kLqcdAyYAaEY"
width="100%" height="680"></iframe>
            </div>
    @endsection

```

## Папка Database

### Файл UsersTableSeeder.php

```

<?php

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class UsersTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('users')->insert([
            'name' => 'user',
            'email' => 'user@gmail.com',
            'password' => bcrypt('12345678'),
        ]);
    }
}

```

### Файл NewsTableSeeder.php

```

<?php

use Illuminate\Database\Seeder;

class NewsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('news')->insert([
            'published' => "1",
            'title' => "Акція 'Життя в стилі еко'",
            'text' => '
14 грудня волонтери СумДУ долучилися до заходу Життя в стилі
еко, що організували співробітники бібліотеки університету.
Волонтери СумДУ виступили з презентацією та розповіли про
проведені акції по очищенню та озелененню міста, що проводили протягом року.
Також показали фотографії та провели міні-вікторину.

```

У такий спосіб, ми мотивуємо школярів до збереження та очищення довкілля, пропагуємо економію невідтворювальних ресурсів та залучаємо всіх оочих до співпраці, - говорить волонтер Катя Богомоллова.

На заході школярі та студенти різних навчальних закладів представляли свої роботи з екологічних матеріалів.

```

',
'photo' => '',
'slug' => '10',
'meta_description' => '',
'meta_keywords' => '',
'created_at' => '2019-01-4 16:22:44',
'updated_at' => null,
]);
DB::table('news')->insert([
'title' => "Новорічна казка для вихованців центру
\"Злагода\"",
'text' => '
21 грудня волонтери СумДУ разом із студентами соціальної
служби організували справжнє свято для вихованців реабілітаційного центру
\"Злагода\".

```

Вони підготували новорічну казку, яку самі й придумали. Там були присутні різні казкові герої: Дід Мороз, Снігурочка, Баба Яга, Кікімора, Леший, Завірюха та навіть Мушкетери, а також символ нового року Свинка.

У кінці заходу вихованці закладу презентували волонтерам дуже ніжний танець, який справив приємне враження. За такий гарний сюрприз вони отримали солодкі подарунки від Діда Мороза.

```

',
'photo' => '',
'slug' => '9',
'meta_description' => '',
'meta_keywords' => '',
'created_at' => '2019-01-14 16:23:50',
'updated_at' => null,
]);
DB::table('news')->insert([
'title' => "Акція \"Ялинка янголів\"",
'text' => '28 грудня відвідали Центр соціально-психічної
реабілітації для дітей області у м.Суми.

```

У рамках акції "Ялинка янголів", що організовує адміністрація ТЦ "Універмаг "Київ", допомогли роздати діткам новорічні подарунки та провели кілька веселих ігор',

```

'photo' => '',
'slug' => '8',
'meta_description' => '',
'meta_keywords' => '',
'created_at' => '2019-01-14 16:25:19',
'updated_at' => null,
]);
DB::table('news')->insert([
'title' => "Дива перед Різдвом",
'text' => 'Перед Різдвом відвідали нашого давнього друга з
Правдинської школи-інтернату. Саша Орленко потрапив у лікарню з опіками
кінцівок - волонтери СумДУ завітали в гості та принесли новорічні подарунки,
кошти на ліки та поспілкувалися з ним.',

```

```

'photo' => '',
'slug' => '7',

```

```

    'meta_description' => '',
    'meta_keywords' => '',
    'created_at' => '2019-01-14 16:27:12',
    'updated_at' => null,
  ]);
  DB::table('news')->insert([
    'published' => "1",
    'title' => "Акція \"Обійми для всіх\"",
    'text' => '

```

21 січня відзначається Міжнародний день обіймів. У цей день по всьому світу люди намагаються обійняти як можна більше людей, навіть незнайомих. Це свято прийшло до нас із Сполучених Штатів Америки, і швидко поширилось серед молоді. Волонтери СумДУ не залишились осторонь і з усією душевною добротою та гарним настроєм вийшли на центральні вулиці міста.

\"Ми мали на меті поділитися емоціями з незнайомцями та загартуватися, - повідомила волонтер Вікторія Гичкаленко. - Адже доброзичливі обійми підвищують імунітет\". Обійшли декілька вулиць, обійняли близько 300 осіб і роздали понад 100 цукерок.

У рамках акції усі, від дітей до людей похилого віку, могли отримати смаколик, обмінявши усмішку на цукерку. Без виключень, волонтери обіймали усіх, хто траплявся на шляху.

Волонтери СумДУ вбачають проблему суспільства в недоброзичливому ставленні та недовірі людей до один одного.

\"Маленькими кроками ми намагаємося змінити ситуацію та показати оточенню, що добрі люди бувають, що набагато краще жити у довірі та щирості\", - говорить волонтер Аня Тарануха.

Сьогодні акція набула нових масштабів, адже представники волонтерського руху СумДУ провели паралельно акцію в Києві, за що дякуємо Єгору Ганжі. Він підходив до перехожих, аби обмінятися з ними приємними словами, а за гарний настрій дарував цукерки.

Також у цей день відвідали будинок дитини на вул. Герасима-Кондратьєва, передали маленьким діткам, зібрані раніше, м'які іграшки та декілька упаковок підгузків, придбання яких проспонсорував випускник СумДУ Винниченко Богдан.

```

    ',
    'photo' => '',
    'slug' => '6',
    'meta_description' => '',
    'meta_keywords' => '',
    'created_at' => '2019-01-22 10:28:55',
    'updated_at' => null,
  ]);
  DB::table('news')->insert([
    'published' => "1",
    'title' => "«Допомога чотирилапим»",
    'text' => '

```

26 січня волонтери СумДУ вкотре відвідали центр перетримки тварин. До вже сформованої групи долучилися і нові студенти-волонтери, які із задоволенням вигуляли більше 20 собачок. Також силами студентів була зібрана сума, на яку закупили м'яса та кілька поводків. Хлопці ж взяли до рук лопати та допомогли волонтерам центру розкидати купу снігу.

«Я ходжу сюди просто так, інколи без акції, а чисто для себе, - говорить волонтер Наташа Петренко. - Мені подобається гуляти з собаками, у кожній своїй

неповторний характер, абсолютно унікальна зовнішність, але всі вони хочуть любові. А я їй можу дати».

```

',
'photo' => '',
'slug' => '5',
'meta_description' => '',
'meta_keywords' => '',
'created_at' => '2019-01-28 16:39:05',
'updated_at' => null,
]);
DB::table('news')->insert([
'published' => "1",
'title' => "Кращі студенти СумДУ на екскурсії у ВРУ",
'text' => '

```

5 лютого волонтери Сумського державного університету разом із студентами кафедри психології, політології та соціокультурних технологій побували у Києві.

Найактивніші студенти-волонтери та студенти, які пов'язані з євроінтеграційними процесами отримали можливість відвідати Верховну Раду України.

Екскурсію було організовано нардепом України Олегом Медуницею. Студентів провели поверхами будівлі, дали змогу відчувати себе депутатами. Показали і синьо-жовтий прапор, який був внесений депутатами до сесійної зали Верховної Ради у 1991 році, історичні картини, які прикрашають стіни приміщення, балкони, звідки видно робочі місця нардепів, кулуари та багато іншого. Вразила студентів й історія самої будівлі та сесійної зали. Дуже сподобалася пропозиція зблизька роздивитися місця законотворців та уявити себе на їх місці. Побачене справило сильне враження.

«Сама поїздка захопила мене ще до того, як почалась. - Розповідає студент-волонтер факультету ЕліТ Стас Федорчук. - Коли ми прибули на Хрещатик, нас зустрів помічник-консультант народного депутата Павло Дацків (який навчався колись на тій спеціальності, що і я зараз). Він супроводжував нас до самої Верховної Ради. У ВРУ було дуже круто: прекрасний екскурсовод, привітні депутати та загальна атмосфера. Я запам'ятаю це надовго!».

Окрім екскурсії Верховною Радою студенти відвідали найголовнішу християнську святиню Східної Європи - Софійський собор. Він є пам'яткою української архітектури, яка залишилася ще з часів Київської Русі, тому молоді було надзвичайно цікаво дізнатися про факти найстарішої споруди Києва.

Екскурсовод розповіла про історію заснування собору та зміни, які відбувалися протягом довгих років, показала мозаїки на його величних стінах. Також там містяться і фрески, які уціліли на сьогоднішній день. Ще однією цікавинкою став саркофаг Ярослава Мудрого. Також у Софії Київській є унікальне панно Богородиці, викладене з 15 тисяч писанок. Після розповідей екскурсовода кожен мав час, щоб добре розгледіти таємничі куточки музею та подивитися й інші будівлі на території собору, зокрема галереї.

«Не дивлячись на те, що я була в Києві декілька разів, мені вдалося дізнатися багато чого цікавого та нового. Я пережила неймовірні емоції», - поділилася враженнями волонтер Аня Тарануха.

```

',
'photo' => '',
'slug' => '4',

```

```

        'meta_description' => '',
        'meta_keywords' => '',
        'created_at' => '2019-02-11 11:55:17',
        'updated_at' => null,
    ]);
    DB::table('news')->insert([
        'published' => "1",
        'title' => "Благодійний «Общажник» зібрав сотню людей",
        'text' => '

```

7 лютого Волонтери СумДУ організували найатмосфернішу подію цієї зими - благодійний «Общажник». Метою заходу став збір коштів Олені Багі, дівчинці, яка лікується у Сумському гематологічному відділенні. На музичний вечір прийшло близько сотні людей, що дало змогу зібрати 2170 грн. «Ми в праві впливати на долі інших людей, то чому б не допомогти тим, кому це необхідно? Тим паче, поєднуючи приємне з корисним», - коментує організатор заходу Анастасія Вихрова.

«Общажник» був традиційним, але експериментальним. Цей івент створений для того, аби познайомити молодь міста із новими музикантами, про яких ми не чули раніше. «Зазвичай, це була гра на гітарі, часто імпрровізація. Та цього разу можна було почути мелодію скрипки та сопілки», - повідомила соорганізатор заходу Вікторія Гичкаленко.

На вечорі свої таланти демонстрували 7 учасників. Серед них були сопілкар Тетяна Шатілова, яка веде свої курси гри на сопілці та скрипачка Вероніка Бабич, яка займається музикою уже 8 років. З власними піснями виступив і організатор фестивалю «Sumy Rock Fest» Володимир Луговий. Також свою творчість презентували молоді дуети «Ноябрь-17», «The last intention» та «INUA», які виконували кавери на сучасних музикантів.

Хедлайнерами заходу були хлопці із групи «DEINDE». Соліст гурту Семен, розповів, що це вже третій «Общажник», на якому вони виступають, а ще, зовсім скоро вийде їх перший альбом «Хвилі».

Лампова атмосфера, килими, гарячий чай та печиво створили домашній затишок та комфорт. У теплому колі зовсім різних людей запанувала єдність - єдність, заради спільної мети!

```

        ',
        'photo' => '',
        'slug' => '3',
        'meta_description' => '',
        'meta_keywords' => '',
        'created_at' => '2019-02-11 13:09:34',
        'updated_at' => null,
    ]);
    DB::table('news')->insert([
        'published' => "1",
        'title' => "Волонтери СумДУ влаштували флешмоб до Дня
закоханих",

```

'text' => '14 лютого волонтери СумДУ спільно з учасниками фольклорно-краєзнавчого гуртка «Краяни» організували масштабний флешмоб.

На площі біля центрального корпусу зібралося понад 100 студентів, представників інститутів та факультетів університету,

які об'єдналися у велике серце. Таким чином вони показали свою любов до навчального закладу у день, коли зізнаються у коханні. Кожен, хто долучився до заходу,

отримав червону кульку та сертифікат на 100 гривень від спонсору «Компсервіс». Під час флешмобу грала романтична музика,



очочі мали можливість зробити чудові фото. Окрім цього продавалися марципани, що були виготовлені власноруч студенткою Медичного інституту

Надією Лебедевою. Виручені кошти у сумі 500 гривень волонтери перерахувати на лікування онкохворій дівчинці із м.Суми Адріані Денисенко.

До організації тематичного заходу приєдналися і волонтери студентської соціальної служби. Вони відвідали відділення денного перебування \ 'Злагода\ '

КУ «Сумського міського територіального центру соціального обслуговування \ 'Берегиня\ '\ ' та разом із його вихованцями виготовили привітальні

листівки до Дня Святого Валентина! Усі залишилися задоволеними та з подарунками.

Дякуємо усім студентам інститутів та факультетів СумДУ, які долучилися до цього дійства.

Саме завдяки вам ми показали, що Сумський державний університет це одна велика родина!',

```
'photo' => '',
'slug' => '2',
'meta_description' => '',
'meta_keywords' => '',
'created_at' => '2019-02-15 15:21:18',
'updated_at' => null,
```

```
]);
```

```
DB::table('news')->insert([
```

```
'published' => "1",
```

```
'title' => "Волонтери СумДУ завітали до Центру соціально-психологічної реабілітації дітей області",
```

```
'text' => '16 лютого волонтери СумДУ відвідали Центр соціально-психічної реабілітації для дітей області, візити до якого вже стали доброю традицією.
```

Під час відвідин для дітлахів провели майстер-клас з малювання звірів за допомогою цікавої техніки, розповіли декілька лайфхаків та просто гарно провели суботній день. Вихованці щиро цікавилися справами волонтерів, а студенти у свою чергу дізнавалися про вподобання та захоплення дітлахів. Старші діти запитували про навчання в університеті та як обирати професію. По закінченню майстер-класу дітлахи почали робити власні вітальні листівки.

Родзинкою заходу стало привітання одного з вихованців центру з днем народження. 8-ми річному Стасу подарували іграшку-трансформер, якою одразу ж зацікавились усі хлопці.

```
',
'photo' => '',
'slug' => '1',
'meta_description' => '',
'meta_keywords' => '',
'created_at' => '2019-02-18 15:30:07',
'updated_at' => null,
```

```
]);
```

```
}
```

```
}
```

Файл BlocksTableSeeder.php

<?php

```

use Illuminate\Database\Seeder;

class BlocksTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('main_blocks')->insert([
            'title' => 'Стипендіальне забезпечення',
            'text' => 'Уряд затвердив новий Порядок
            призначення і виплати стипендій
            учням професійно-технічних,
            студентам (курсантам) вищих
            державних та комунальних
            навчальних закладів...',
            'link' => 'http://socstor-test.sumdu.edu.ua/scholarship',
            'type' => '0'
        ]);
        DB::table('main_blocks')->insert([
            'title' => 'Фотоальбоми',
            'text' => 'Уряд затвердив новий Порядок
            призначення і виплати стипендій
            учням професійно-технічних,
            студентам (курсантам) вищих
            державних та комунальних
            навчальних закладів...',
            'link' => 'http://socstor-test.sumdu.edu.ua/albums',
            'type' => '0'
        ]);
        DB::table('main_blocks')->insert([
            'title' => 'Карта добрих справ',
            'text' => 'Карта добрих справ - це
            наглядний приклад того, що наші
            волонтери дійсно допомагають
            усій Україні, від Львова до
            Луганська. Ми ділимося добром зі
            школами, дитсадками...',
            'link' => 'http://socstor-test.sumdu.edu.ua/map',
            'type' => '0'
        ]);
        DB::table('main_blocks')->insert([
            'title' => 'Розміри стипендій студентів вищих навчальних
            закладів',
            'text' => 'Розмір мінімальної ординарної (звичайної) академічної
            стипендії:

            • для студентів вищих навчальних закладів III - IV рівня
            акредитації, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр",
            "спеціаліст" або "магістр", крім тих, що навчаються за спеціальностями
            (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному
            розмірі, - 1100 гривень на місяць;

            • для студентів вищих навчальних закладів III - IV рівня
            акредитації, наукових установ, які навчаються за освітньо-кваліфікаційним
            рівнем "бакалавр", "спеціаліст" або "магістр", що навчаються за

```

спеціальностями (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному розмірі, - 1400 гривень на місяць.

Розмір соціальної стипендії для студентів вищих навчальних закладів з числа дітей-сиріт та дітей, позбавлених батьківського піклування, осіб з їх числа, а також студентів, які в період навчання у віці від 18 до 23 років залишилися без батьків:

```

        • становить - 2000 гривень на місяць. ',
        'type' => '1'
    ]);
    DB::table('main_blocks')->insert([
        'title' => 'Перелік спеціальностей, для яких встановлюються академічні стипендії у підвищеному розмірі в СумДУ',
        'text' => 'Розмір мінімальної ординарної (звичайної) академічної стипендії: • для студентів вищих навчальних закладів III - IV рівня акредитації, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр", "спеціаліст" або "магістр", крім тих, що навчаються за спеціальностями (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному розмірі, - 1100 гривень на місяць; • для студентів вищих навчальних закладів III - IV рівня акредитації, наукових установ, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр", "спеціаліст" або "магістр", що навчаються за спеціальностями (спеціалізаціями), для яких встановлюються ',
        'type' => '1'
    ]);
    DB::table('main_blocks')->insert([
        'title' => 'Нормативні документи згідно яких відбувається призначення академічних та соціальних стипендій',
        'text' => '• для студентів вищих навчальних закладів III - IV рівня акредитації, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр", "спеціаліст" або "магістр", крім тих, що навчаються за спеціальностями (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному розмірі, - 1100 гривень на місяць;

        • для студентів вищих навчальних закладів III - IV рівня акредитації, наукових установ, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр", "спеціаліст" або "магістр", що навчаються за спеціальностями (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному розмірі, - 1400 гривень на місяць.

        • для студентів вищих навчальних закладів III - IV рівня акредитації, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр", "спеціаліст" або "магістр", крім тих, що навчаються за спеціальностями (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному розмірі, - 1100 гривень на місяць;

        • для студентів вищих навчальних закладів III - IV рівня акредитації, наукових установ, які навчаються за освітньо-кваліфікаційним рівнем "бакалавр", "спеціаліст" або "магістр", що навчаються за спеціальностями (спеціалізаціями), для яких встановлюються академічні стипендії у підвищеному розмірі, - 1400 гривень на місяць.',
        'type' => '1'
    ]);
    });
}
}

```

Папка App

Файл Kernel.php (папка Http)

```

<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your
application.
     *
     * @var array
     */
    protected $middleware = [
        \App\Http\Middleware\CheckForMaintenanceMode::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,

\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \App\Http\Middleware\TrustProxies::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array
     */
    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,

\Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            //
\Illuminate\Session\Middleware\AuthenticateSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \App\Http\Middleware\VerifyCsrfToken::class,
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],

        'api' => [
            'throttle:60,1',
            'bindings',
        ],
    ];

    /**
     * The application's route middleware.
     *
     * These middleware may be assigned to groups or used individually.
     *
     * @var array
     */
    protected $routeMiddleware = [
        'auth' => \App\Http\Middleware\Authenticate::class,
    ];
}

```

```

        'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
        'bindings' =>
\Illuminate\Routing\Middleware\SubstituteBindings::class,
        'cache.headers' =>
\Illuminate\Http\Middleware\SetCacheHeaders::class,
        'can' => \Illuminate\Auth\Middleware\Authorize::class,
        'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
        'signed' =>
\Illuminate\Routing\Middleware\ValidateSignature::class,
        'throttle' =>
\Illuminate\Routing\Middleware\ThrottleRequests::class,
        'verified' =>
\Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    ];

    /**
     * The priority-sorted list of middleware.
     *
     * This forces non-global middleware to always be in the given
order.
     *
     * @var array
     */
    protected $middlewarePriority = [
        \Illuminate\Session\Middleware\StartSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\Authenticate::class,
        \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
        \Illuminate\Auth\Middleware\Authorize::class,
    ];
}

```

### Файл KeyWords.php (папка Rules)

```

<?php

namespace App\Rules;

use Illuminate\Contracts\Validation\Rule;

class KeyWords implements Rule
{
    /**
     * Create a new rule instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Determine if the validation rule passes.
     *
     * @param string $attribute

```

```

* @param mixed $value
* @return bool
*/
public function passes($attribute, $value)
{
    $regex = "/^(([a-яіігёА-ЯІІГЁ]| [a-zA-Z]| [0-9])+(\, \ ))*([a-яіігёА-
ЯІІГЁ]| [a-zA-Z]| [0-9])+$ /ui";

    return preg_match($regex, $value);
}

/**
 * Get the validation error message.
 *
 * @return string
 */
public function message()
{
    return 'Ключові слова перелічити через кому (слово1, слово2, слово3,
...).' ;
}
}

```

## Файл Kernel.php (папка Console)

```

<?php

namespace App\Console;

use Illuminate\Console\Scheduling\Schedule;
use Illuminate\Foundation\Console\Kernel as ConsoleKernel;

class Kernel extends ConsoleKernel
{
    /**
     * The Artisan commands provided by your application.
     *
     * @var array
     */
    protected $commands = [
        //
    ];

    /**
     * Define the application's command schedule.
     *
     * @param \Illuminate\Console\Scheduling\Schedule $schedule
     * @return void
     */
    protected function schedule(Schedule $schedule)
    {
        // $schedule->command('inspire')
        //             ->hourly();
    }

    /**

```

```

    * Register the commands for the application.
    *
    * @return void
    */
protected function commands()
{
    $this->load(__DIR__.'/Commands');

    require base_path('routes/console.php');
}
}

```

## Файл Handler.php (папка Exceptions)

```

<?php

namespace App\Exceptions;

use Exception;
use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;
use Illuminate\Auth\AuthenticationException;

class Handler extends ExceptionHandler
{
    /**
     * A list of the exception types that are not reported.
     *
     * @var array
     */
    protected $dontReport = [
        //
    ];

    /**
     * A list of the inputs that are never flashed for validation
exceptions.
     *
     * @var array
     */
    protected $dontFlash = [
        'password',
        'password_confirmation',
    ];

    /**
     * Report or log an exception.
     *
     * @param \Exception $exception
     * @return void
     */
    public function report(Exception $exception)
    {
        parent::report($exception);
    }

    /**

```

```

* Render an exception into an HTTP response.
*
* @param \Illuminate\Http\Request $request
* @param \Exception $exception
* @return \Illuminate\Http\Response
*/
public function render($request, Exception $exception)
{
    // if ($exception instanceof AuthenticationException) {

    //     $guard = array_get($exception->guards(), 0);
    //     switch ($guard) {
    //         case 'admin':
    //             return redirect()->guest(route('admin.login') );
    //             break;

    //         default:
    //             return redirect()->guest(route('login'));
    //             break;
    //     }
    // }
    return parent::render($request, $exception);
}

/**
 * Convert an authentication exception into an unauthenticated
response.
 *
 * @param \Illuminate\Http\Request $request
 * @param \Illuminate\Auth\AuthenticationException $exception
 * @return \Illuminate\Http\Response
 */
protected function unauthenticated($request, AuthenticationException
$exception)
{
    if ($request->expectsJson()) {
        return response(['error' => 'Unauthenticated.'],
401);
    }

    $guard = array_get($exception->guards(), 0);
    switch ($guard) {
        case 'admin':
            $login = 'admin.login';
            break;

        default:
            $login = 'login';
            break;
    }
    return redirect()->guest(route($login));
}
}

```

## Файл Slider.php

<?php



```

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Str;

class Slider extends Model
{
    protected $fillable = ['title', 'image', 'description', 'link',
'type'];

    public function setSlugAttribute($value)
    {
        $this->attributes['link'] = Str::link( mb_substr($this->title,
0, 40));
    }

    public function scopeLastNew($query, $count)
    {
        return $query->orderBy('created_at', 'desc')->take($count)-
>get();
    }

    public function scopeTypeSlider($query)
    {
        return $query->where('type', 1);
    }

    public function scopeTypePartners($query)
    {
        return $query->where('type', 2);
    }

    public static function parsePhoto($data)
    {
        // path to store
        $path = 'uploads/';

        $imageInfo = explode(";base64,", $data);
        $imgExt = str_replace('data:image/', '', $imageInfo[0]);
        $image = str_replace(' ', '+', $imageInfo[1]);
        $imageName = "slider-" . time() . "." . $imgExt;

        return [
            'path' => $path . $imageName,
            'base64' => $image
        ];
    }

    public function getPath()
    {
        $path = null;
        if ($this->image) {
            $path = asset('/storage/' . $this->image);
        } else {
            $path = 'https://via.placeholder.com/700';
        }

        return $path;
    }

    // public function getType() {

```

```

        //      $type = null;

        //      switch ($this->type) {
        //          case 0:
        //              $type = 'Головна';
        //              break;
        //          case 1:
        //              $type = 'Стипендіальне забезпечення';
        //              break;
        //      }

        //      return $type;
        //  }

    }

```

## Папка Bootstrap

### Файл services.php

```

<?php return array (
    'providers' =>
    array (
        0 => 'Illuminate\Auth\AuthServiceProvider',
        1 => 'Illuminate\Broadcasting\BroadcastServiceProvider',
        2 => 'Illuminate\Bus\BusServiceProvider',
        3 => 'Illuminate\Cache\CacheServiceProvider',
        4 => 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
        5 => 'Illuminate\Cookie\CookieServiceProvider',
        6 => 'Illuminate\Database\DatabaseServiceProvider',
        7 => 'Illuminate\Encryption\EncryptionServiceProvider',
        8 => 'Illuminate\Filesystem\FilesystemServiceProvider',
        9 => 'Illuminate\Foundation\Providers\FoundationServiceProvider',
        10 => 'Illuminate\Hashing\HashServiceProvider',
        11 => 'Illuminate\Mail\MailServiceProvider',
        12 => 'Illuminate\Notifications\NotificationServiceProvider',
        13 => 'Illuminate\Pagination\PaginationServiceProvider',
        14 => 'Illuminate\Pipeline\PipelineServiceProvider',
        15 => 'Illuminate\Queue\QueueServiceProvider',
        16 => 'Illuminate\Redis\RedisServiceProvider',
        17 => 'Illuminate\Auth\Passwords>PasswordResetServiceProvider',
        18 => 'Illuminate\Session\SessionServiceProvider',
        19 => 'Illuminate\Translation\TranslationServiceProvider',
        20 => 'Illuminate\Validation\ValidationServiceProvider',
        21 => 'Illuminate\View\ViewServiceProvider',
        22 => 'BeyondCode\DumpServer\DumpServerServiceProvider',
        23 => 'Fideloper\Proxy\TrustedProxyServiceProvider',
        24 => 'Intervention\Image\ImageServiceProvider',
        25 => 'Jenssegers>Date\DateServiceProvider',
        26 => 'Laravel\Tinker\TinkerServiceProvider',
        27 => 'Carbon\Laravel\ServiceProvider',
        28 =>
        'NunoMaduro\Collision\Adapters\Laravel\CollisionServiceProvider',
        29 => 'App\Providers\AppServiceProvider',
        30 => 'App\Providers\AuthServiceProvider',
        31 => 'App\Providers\EventServiceProvider',
        32 => 'App\Providers\RouteServiceProvider',
    )
)

```

```

33 => 'Intervention\\Image\\ImageServiceProvider',
34 => 'Unisharp\\Ckeditor\\ServiceProvider',
35 => 'Jenssegers\\Date\\DateServiceProvider',
36 => 'App\\Providers\\SocServiceProvider',
),
'eager' =>
array (
    0 => 'Illuminate\\Auth\\AuthServiceServiceProvider',
    1 => 'Illuminate\\Cookie\\CookieServiceProvider',
    2 => 'Illuminate\\Database\\DatabaseServiceProvider',
    3 => 'Illuminate\\Encryption\\EncryptionServiceProvider',
    4 => 'Illuminate\\Filesystem\\FilesystemServiceProvider',
    5 => 'Illuminate\\Foundation\\Providers\\FoundationServiceProvider',
    6 => 'Illuminate\\Notifications\\NotificationServiceProvider',
    7 => 'Illuminate\\Pagination\\PaginationServiceProvider',
    8 => 'Illuminate\\Session\\SessionServiceProvider',
    9 => 'Illuminate\\View\\ViewServiceProvider',
    10 => 'BeyondCode\\DumpServer\\DumpServerServiceProvider',
    11 => 'Fideloper\\Proxy\\TrustedProxyServiceProvider',
    12 => 'Intervention\\Image\\ImageServiceProvider',
    13 => 'Jenssegers\\Date\\DateServiceProvider',
    14 => 'Carbon\\Laravel\\ServiceProvider',
    15 => 'App\\Providers\\AppServiceProvider',
    16 => 'App\\Providers\\AuthServiceServiceProvider',
    17 => 'App\\Providers\\EventServiceProvider',
    18 => 'App\\Providers\\RouteServiceProvider',
    19 => 'Intervention\\Image\\ImageServiceProvider',
    20 => 'Unisharp\\Ckeditor\\ServiceProvider',
    21 => 'Jenssegers\\Date\\DateServiceProvider',
    22 => 'App\\Providers\\SocServiceProvider',
),
'deferred' =>
array (
    'Illuminate\\Broadcasting\\BroadcastManager' =>
'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Contracts\\Broadcasting\\Factory' =>
'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Contracts\\Broadcasting\\Broadcaster' =>
'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Bus\\Dispatcher' => 'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Contracts\\Bus\\Dispatcher' =>
'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Contracts\\Bus\\QueueingDispatcher' =>
'Illuminate\\Bus\\BusServiceProvider',
    'cache' => 'Illuminate\\Cache\\CacheServiceProvider',
    'cache.store' => 'Illuminate\\Cache\\CacheServiceProvider',
    'memcached.connector' => 'Illuminate\\Cache\\CacheServiceProvider',
    'command.cache.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.cache.forget' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.clear-compiled' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.auth.resets.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.config.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.config.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.down' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.environment' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

'command.event.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.event.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.event.list' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.key.generate' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate.fresh' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate.install' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate.refresh' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate.reset' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate.rollback' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.migrate.status' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.optimize' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.optimize.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.package.discover' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.preset' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.failed' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.flush' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.forget' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.listen' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.restart' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.retry' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.queue.work' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.route.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.route.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.route.list' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.seed' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleFinishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'Illuminate\\Console\\Scheduling\\ScheduleRunCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.storage.link' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.up' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'command.view.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

    'command.view.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.app.name' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.auth.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.cache.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.channel.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.console.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.controller.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.event.generate' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.event.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.exception.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.factory.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.job.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.listener.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.mail.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.middleware.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.migrate.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.model.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.notification.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.notification.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.observer.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.policy.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.provider.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.queue.failed-table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.queue.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.request.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.resource.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.rule.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.seeder.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.session.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.serve' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
  'command.test.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

```

    'command.vendor.publish' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'migrator' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'migration.repository' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'migration.creator' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'composer' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
'hash' => 'Illuminate\\Hashing\\HashServiceProvider',
'hash.driver' => 'Illuminate\\Hashing\\HashServiceProvider',
'mailer' => 'Illuminate\\Mail\\MailServiceProvider',
'swift.mailer' => 'Illuminate\\Mail\\MailServiceProvider',
'swift.transport' => 'Illuminate\\Mail\\MailServiceProvider',
'Illuminate\\Mail\\Markdown' => 'Illuminate\\Mail\\MailServiceProvider',
'Illuminate\\Contracts\\Pipeline\\Hub' =>
'Illuminate\\Pipeline\\PipelineServiceProvider',
'queue' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.worker' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.listener' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.failer' => 'Illuminate\\Queue\\QueueServiceProvider',
'queue.connection' => 'Illuminate\\Queue\\QueueServiceProvider',
'redis' => 'Illuminate\\Redis\\RedisServiceProvider',
'redis.connection' => 'Illuminate\\Redis\\RedisServiceProvider',
'auth.password' =>
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
'auth.password.broker' =>
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
'translator' => 'Illuminate\\Translation\\TranslationServiceProvider',
'translation.loader' =>
'Illuminate\\Translation\\TranslationServiceProvider',
'validator' => 'Illuminate\\Validation\\ValidationServiceProvider',
'validation.presence' =>
'Illuminate\\Validation\\ValidationServiceProvider',
'command.tinker' => 'Laravel\\Tinker\\TinkerServiceProvider',
'NunoMaduro\\Collision\\Contracts\\Provider' =>
'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',
),
'when' =>
array (
    'Illuminate\\Broadcasting\\BroadcastServiceProvider' =>
array (
    ),
    'Illuminate\\Bus\\BusServiceProvider' =>
array (
    ),
    'Illuminate\\Cache\\CacheServiceProvider' =>
array (
    ),
    'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider' =>
array (
    ),
    'Illuminate\\Hashing\\HashServiceProvider' =>
array (
    ),
    'Illuminate\\Mail\\MailServiceProvider' =>
array (
    ),
    'Illuminate\\Pipeline\\PipelineServiceProvider' =>
array (
    ),
    'Illuminate\\Queue\\QueueServiceProvider' =>

```

```
array (
),
'Illuminate\\Redis\\RedisServiceProvider' =>
array (
),
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider' =>
array (
),
'Illuminate\\Translation\\TranslationServiceProvider' =>
array (
),
'Illuminate\\Validation\\ValidationServiceProvider' =>
array (
),
'Laravel\\Tinker\\TinkerServiceProvider' =>
array (
),
'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider' =>
array (
),
),
);
```

## ДОДАТОК Б

### Програмний код проєкту бази даних та міграцій

Папка Database

Файл 2014\_10\_12\_100000\_create\_password\_resets\_table (папка

migrations)

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('password_resets', function (Blueprint $table) {
            $table->string('email')->index();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('password_resets');
    }
}
```

Файл 2014\_10\_12\_000000\_create\_users\_table

```
class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }
}
```



```

        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}

```

### Файл 2019\_05\_31\_104127\_create\_admins\_table

```

<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateAdminsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('admins', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
}

```

```

public function down()
{
    Schema::dropIfExists('admins');
}
}

```

## Файл phpunit.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<phpunit backupGlobals="false"
    backupStaticAttributes="false"
    bootstrap="vendor/autoload.php"
    colors="true"
    convertErrorsToExceptions="true"
    convertNoticesToExceptions="true"
    convertWarningsToExceptions="true"
    processIsolation="false"
    stopOnFailure="false">
    <testsuites>
        <testsuite name="Unit">
            <directory suffix="Test.php">./tests/Unit</directory>
        </testsuite>

        <testsuite name="Feature">
            <directory suffix="Test.php">./tests/Feature</directory>
        </testsuite>
    </testsuites>
    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">./app</directory>
        </whitelist>
    </filter>
    <php>
        <server name="APP_ENV" value="testing"/>
        <server name="BCRYPT_ROUNDS" value="4"/>
        <server name="CACHE_DRIVER" value="array"/>
        <server name="MAIL_DRIVER" value="array"/>
        <server name="QUEUE_CONNECTION" value="sync"/>
        <server name="SESSION_DRIVER" value="array"/>
    </php>
</phpunit>

```

## Папка config

### Файл database.php

```

<?php

use Illuminate\Support\Str;

return [

    /*
    |-----
    --
    | Default Database Connection Name

```

```

|-----|
--
|
| Here you may specify which of the database connections below you wish
| to use as your default connection for all database work. Of course
| you may use many connections at once using the Database library.
|
*/
'default' => env('DB_CONNECTION', 'mysql'),
/*
|-----|
--
| Database Connections
|-----|
--
|
| Here are each of the database connections setup for your application.
| Of course, examples of configuring each database platform that is
| supported by Laravel is shown below to make development simple.
|
|
| All database work in Laravel is done through the PHP PDO facilities
| so make sure you have the driver for your particular database of
| choice installed on your machine before you begin development.
|
*/
'connections' => [
    'sqlite' => [
        'driver' => 'sqlite',
        'url' => env('DATABASE_URL'),
        'database' => env('DB_DATABASE',
database_path('database.sqlite')),
        'prefix' => '',
        'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
    ],
    'mysql' => [
        'driver' => 'mysql',
        'url' => env('DATABASE_URL'),
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'prefix_indexes' => true,
        'strict' => true,
        'engine' => null,
        'options' => extension_loaded('pdo_mysql') ? array_filter([
            PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
        ]) : [],
    ],
    'pgsql' => [
        'driver' => 'pgsql',
        'url' => env('DATABASE_URL'),

```

```

        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '5432'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'charset' => 'utf8',
        'prefix' => '',
        'prefix_indexes' => true,
        'schema' => 'public',
        'sslmode' => 'prefer',
    ],

    'sqlsrv' => [
        'driver' => 'sqlsrv',
        'url' => env('DATABASE_URL'),
        'host' => env('DB_HOST', 'localhost'),
        'port' => env('DB_PORT', '1433'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'charset' => 'utf8',
        'prefix' => '',
        'prefix_indexes' => true,
    ],

],

/*
|-----|
-- | Migration Repository Table
|-----|
-- |
| This table keeps track of all the migrations that have already run for
| your application. Using this information, we can determine which of
| the migrations on disk haven't actually been run in the database.
|
*/

'migrations' => 'migrations',

/*
|-----|
-- | Redis Databases
|-----|
-- |
| Redis is an open source, fast, and advanced key-value store that also
| provides a richer body of commands than a typical key-value system
| such as APC or Memcached. Laravel makes it easy to dig right in.
|
*/

'redis' => [

    'client' => env('REDIS_CLIENT', 'predis'),

    'options' => [
        'cluster' => env('REDIS_CLUSTER', 'predis'),
        'prefix' => Str::slug(env('APP_NAME', 'laravel'),
            '_').'_database_',
    ],

```

```

    ],
    'default' => [
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'password' => env('REDIS_PASSWORD', null),
        'port' => env('REDIS_PORT', 6379),
        'database' => env('REDIS_DB', 0),
    ],
    'cache' => [
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'password' => env('REDIS_PASSWORD', null),
        'port' => env('REDIS_PORT', 6379),
        'database' => env('REDIS_CACHE_DB', 1),
    ],
],
];

```

## Папка tests

### Файл ExampleTest.php

```

<?php
namespace Tests\Unit;

use Tests\TestCase;
use Illuminate\Foundation\Testing\RefreshDatabase;

class ExampleTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testBasicTest()
    {
        $this->assertTrue(true);
    }
}

```

### Файл ExampleTest.php

```

<?php
namespace Tests\Feature;

use Tests\TestCase;
use Illuminate\Foundation\Testing\RefreshDatabase;

class ExampleTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testBasicTest()
    {

```

```

        $response = $this->get('/');

        $response->assertStatus(200);
    }
}

```

### Файл CreatesApplication.php

```

<?php

namespace Tests;

use Illuminate\Contracts\Console\Kernel;

trait CreatesApplication
{
    /**
     * Creates the application.
     *
     * @return \Illuminate\Foundation\Application
     */
    public function createApplication()
    {
        $app = require __DIR__.'/../bootstrap/app.php';

        $app->make(Kernel::class)->bootstrap();

        return $app;
    }
}

```

### Файл TestCase.php

```

<?php

namespace Tests;

use Illuminate\Foundation\Testing\TestCase as BaseTestCase;

abstract class TestCase extends BaseTestCase
{
    use CreatesApplication;
}

```

### Папка routes

#### Файл web.php

```

<?php
use Illuminate\Foundation\Auth\User;
use Intervention\Image\Facades\Image;
use App\Http\Controllers\ImageController;
use App\Http\Controllers\User\AlbumController;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!

```

```

|
*/

Route::group(['prefix'=>'user', 'namespace'=>'User', 'middleware'=>['auth']],
function() {
    Route::get('/', 'DashboardController@dashboard')->name('user.index');
    Route::resource('/news', 'NewsController', ['as'=>'user']);
    Route::post('/news/{news}', 'NewsController@update', ['as'=>'user'])-
>name('user.news.update');
    Route::resource('/slider', 'SliderController', ['as'=>'user']);
    Route::resource('/partners', 'PartnersSliderController', ['as'=>'user']);
    Route::resource('/pages', 'PagesController', ['as' => 'user'])-
>except(['create', 'store', 'destroy']);

    // gallery
    Route::group(['prefix' => 'photo-gallery'], function () {
        Route::post('/{photo_gallery}', 'PhotoGalleryController@update')-
>name('user.photo-gallery.update');
        Route::post('/image/upload-photo',
'PhotoGalleryController@uploadPhoto')->name('user.photo-
gallery.image.upload');
        Route::post('/image/destroy-photo',
'PhotoGalleryController@destroyPhoto')->name('user.photo-
gallery.image.destroy');
    });
    Route::resource('/photo-gallery', 'PhotoGalleryController', ['as' =>
'user']);

    Route::group(['prefix' => 'albumchange'], function() {
        Route::get('/', 'AlbumController@index')->name('albumchange.index');
    });

    Route::group(['prefix' => 'news/image'], function() {
        Route::get('/', 'ImageController@index')->name('image.index');
        Route::post('/upload', 'ImageController@upload')-
>name('image.upload');
        Route::post('/destroy', 'ImageController@destroy')-
>name('image.destroy');
    });

    Route::resource('/blocks', 'BlockController', ['as' => 'user'])-
>except(['create', 'store', 'destroy']);
    Route::resource('/employee', 'EmployeeController', ['as' => 'user']);
});

// Front route
Route::get('/', 'HomeController@index')->name('welcome');

//api for article
Route::get('/loadMore', 'HomeController@loadMore')->name('loadMore');
Route::get('/endList', 'HomeController@endList')->name('endList');
Route::get('/albumsVue', 'HomeController@albumsVue')->name('albumsVue');

// end api

Route::get('/news', 'HomeController@news')->name('news');
Route::get('/news/{slug}', 'HomeController@cur_news')->name('cur_news');
Route::get('/map', 'HomeController@map')->name('map');
Route::get('/volunteer', 'HomeController@volunteer')->name('volunteer');
Route::get('/scholarship', 'HomeController@scholarship')-
>name('scholarship');
Route::get('/albums', 'HomeController@albums')->name('albums');

```

```

Route::get('/album', 'HomeController@album')->name('album');
Route::get('/social', 'HomeController@social')->name('social');
Route::get('/social-photo', 'HomeController@getSocialPhoto')->name('social.photo');
Route::get('/department', 'HomeController@department')->name('department');
Route::get('/search', 'HomeController@search')->name('search');

// End front route

Auth::routes(['register' => false, 'reset' => false]);

// Super Admin
Route::prefix('admin')->group( function () {
    Route::get('/login', 'Auth\AdminLoginController@showLoginForm')->name('admin.login');
    Route::get('/logout', 'Auth\AdminLoginController@logout')->name('admin.logout');
    Route::post('/login', 'Auth\AdminLoginController@login')->name('admin.login.submit');
    Route::resource('/configs', 'Admin\ConfigController', ['as' => 'admin'])->except([
        'create', 'show', 'store', 'destroy'
    ]);
    Route::get('/', 'Admin\AdminController@index')->name('admin.dashboard');
    Route::group(['prefix' => 'user-management', 'namespace' => 'Admin\UserManagement'], function() {
        Route::resource('/users', 'UserController', ['as' => 'admin.user-management'])->except('show');
        Route::resource('/admins', 'AdminController', ['as' => 'admin.user-management'])->except('show');
    });
});

// custom method for user logout
Route::post('user/logout', 'Auth>LoginController@userLogout')->name('user.logout');

```



## ДОДАТОК В

### Папка socstor

#### Файл readme.md

```

#Інструкції
- clone repo
- composer install
- copi or rename .env.example .env config db
- php artisan key:generate
- php artisan storage:link
- php artisan migrate --seed
- npm install
- create dir: uploads, photo, cover in storage/app/public/ and added
permissions
- added permissions to storage/logs/ & storage/framework/sessions/ &
storage/framework/views/

> SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was
too long; max key
length is 767 bytes потрібно прописати
```Schema::defaultStringLength(191);``` в
> app\Providers\AppServiceProvider.php

```

При першій установці прописати `php artisan migrate --seed`

Якщо база даних пише що не вичтачає якихось полів то потрібно набрати cli  
`php artisan migrate:fresh --seed`  
*## Щоб мати можливість зберігати файли та читати їх потрібно виконати команду:*  
php artisan storage:link

#### *## Користувачі*

Щоб додати користувача потрібно в терміналі набрати команду `php artisan db:seed`  
- User data: email: user@gmail.com password: 12345678  
- Admin data: email: admin@gmail.com password: 12345678

#### *## Помилки*

> Якщо сіди не виконуються, то потрібно виконати команду `composer dump-autoload`  
> Якщо додали новий компонент то потрібно також виконати `composer dump-autoload`