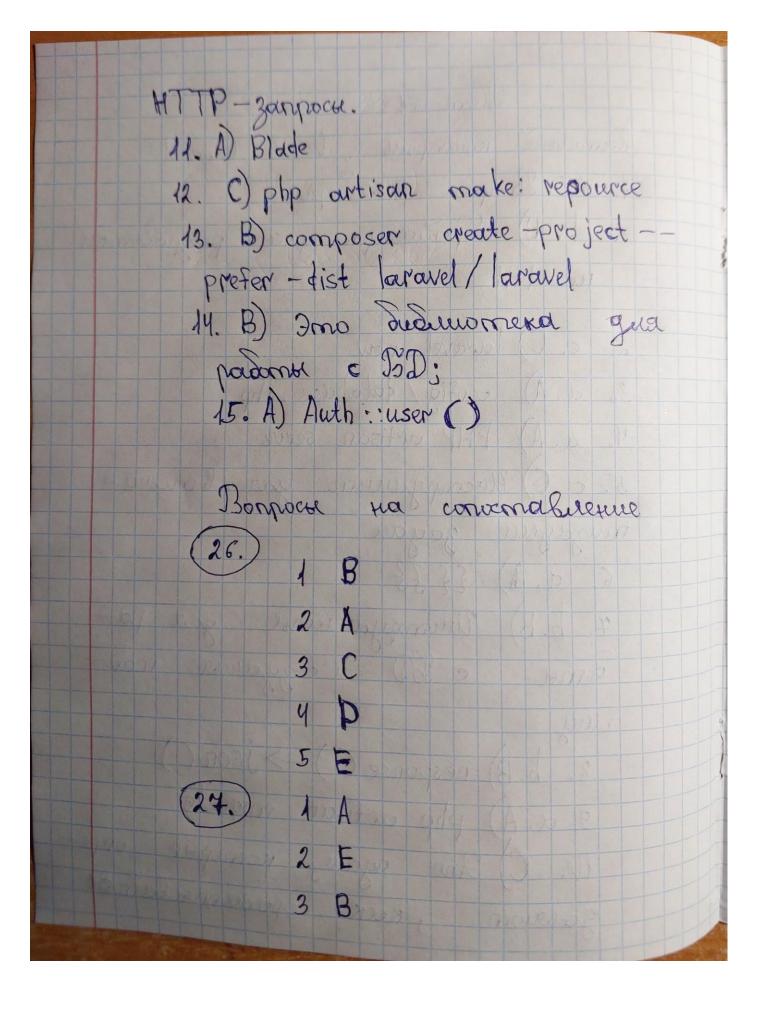
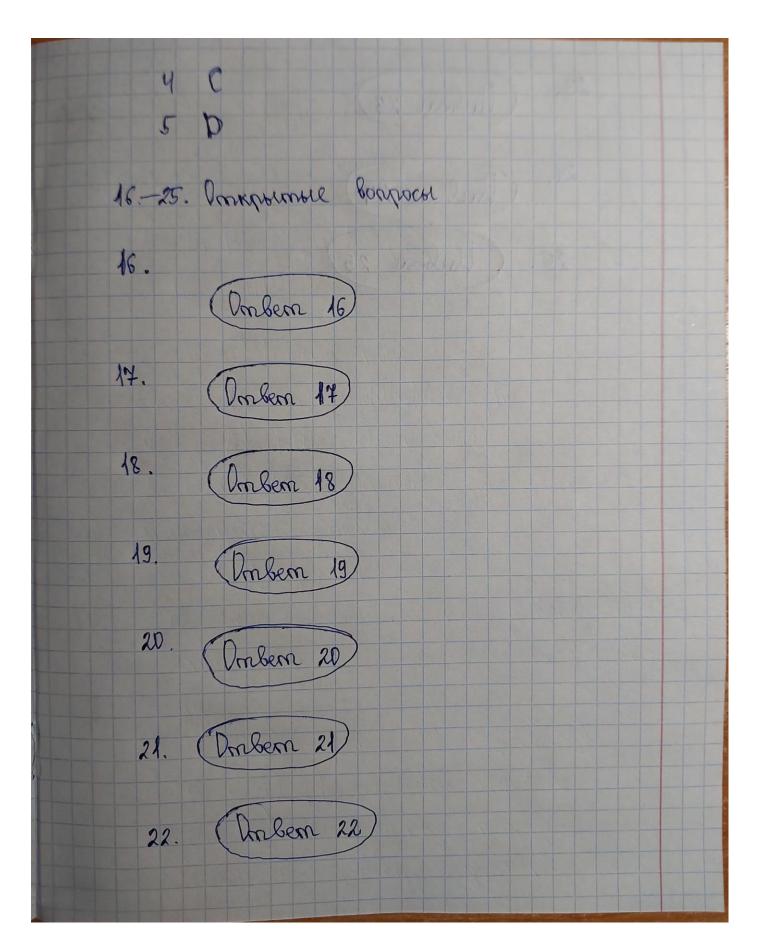
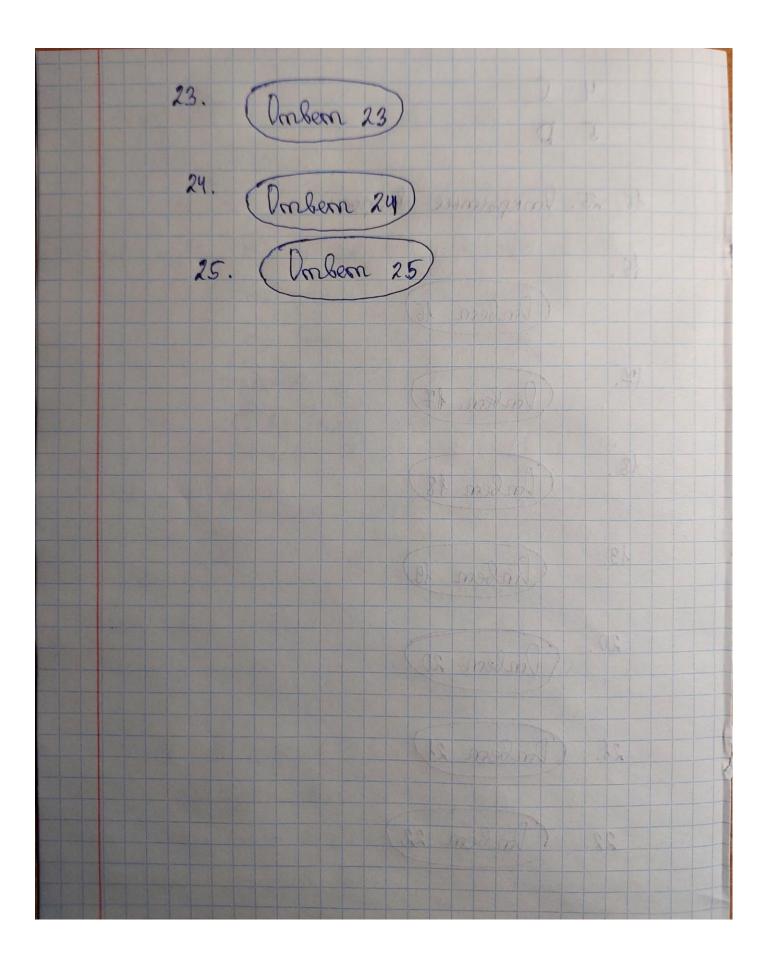
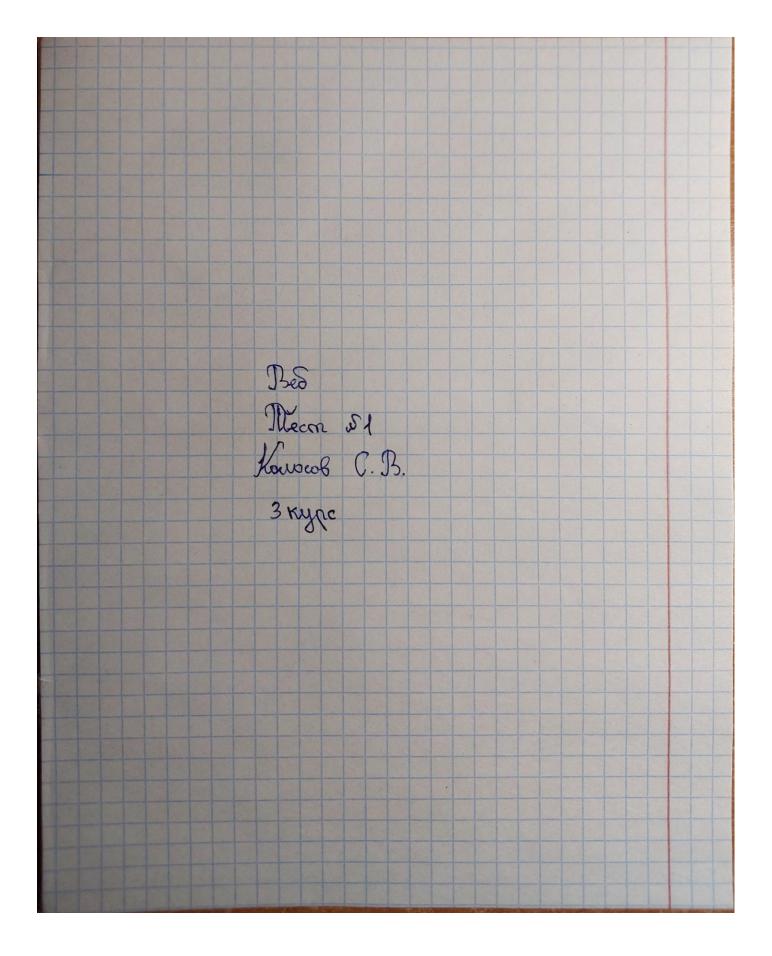


Them w1 Meconobbue Koumpaul ,, Ochobe Laravel" 1.-15. Borgoco e Bendopan 1. b. B) Premubopa gua pazpadoman Ha PHP 2. c. C) lowavel new 3. a. A) config/database. php 4. a. A) PI-IP artisan serve 5. c. C) Miconpymeno que ab manamuzayını zagan 6. a.A) {{ 3}} 4. b.B) Unampequente que pa-Somer e BD u cozganua mais my. 8. b.B) response () -> json () 9. a. A) php artisan migrate 10. C) Ino myone, komopere ongegenerom, kak apadamenbanka









Middleware B Laravel

Middleware в Laravel — это механизм, который позволяет фильтровать HTTP-запросы, выполняемые вашим приложением. Middleware может использоваться для выполнения различных задач, таких как аутентификация, логирование, проверка прав доступа и т.д.

Пример использования middleware:

```
// Создание middleware
php artisan make:middleware CheckAge
```

```
// Регистрация middleware в ядре приложения (app/Http/Kernel.php)

protected $routeMiddleware = [
    'check.age' => \App\Http\Middleware\CheckAge::class,
];

// Применение middleware к маршруту

Route::get('user/profile', function () {
```

(Inchem 16)

})->middleware('check.age');

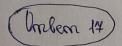
Разница между GET и POST запросами в Laravel

- **GET-запросы** используются для получения данных с сервера. Они кэшируются, могут быть закладками и сохраняются в истории браузера. Данные передаются через URL.
- **POST-запросы** используются для отправки данных на сервер. Они не кэшируются, не могут быть закладками и не сохраняются в истории браузера. Данные передаются в теле запроса.

Пример маршрутов:

Route::get('/user', 'UserController@index');

Route::post('/user', 'UserController@store');



Контроллеры в Laravel

Контроллеры в Laravel используются для группировки логики обработки запросов. Они помогают организовать код и сделать его более читаемым и поддерживаемым.

Пример контроллера:

```
// Создание контроллера

php artisan make:controller UserController

// Пример метода в контроллере

public function index()

{
    $users = User::all();
    return view('users.index', compact('users'));
}
```

Ombem 18)

Связь "один ко многим" в Laravel

Связь "один ко многим" устанавливается, когда одна модель имеет множество связанных моделей.

Пример:

```
// Модель Post
class Post extends Model
{
    public function comments()
    {
        return $this->hasMany(Comment::class);
    }
}

// Модель Comment
class Comment extends Model
{
    public function post()
    {
        return $this->belongsTo(Post::class);
    }
}
```



Зависимость инъекций (Dependency Injection) в Laravel

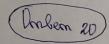
Dependency Injection — это паттерн, который позволяет внедрять зависимости в класс через конструктор или методы. В Laravel это используется для упрощения тестирования и управления зависимостями.

Пример:

```
// Внедрение зависимости в контроллер

public function __construct(UserRepository $users)

{
    $this->users = $users;
}
```



Создание кастомной миграции в Laravel

Миграции используются для управления структурой базы данных.

Пример создания и применения миграции:

```
// Создание миграции
php artisan make:migration create_users_table

// Пример миграции
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamps();
    });
}

// Применение миграции
php artisan migrate
```

Onlen 21)

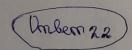
Валидация данных в Laravel

Валидация данных в Laravel может быть выполнена с помощью встроенных правил валидации.

Пример:

```
public function store(Request $request)
{
    $validated = $request->validate([
        'name' => 'required|max:255',
        'email' => 'required|email|unique:users',
    ]);

// Сохранение данных
```



События и слушатели (Events & Listeners) в Laravel

События и слушатели позволяют реализовать асинхронную обработку задач.

Пример:

```
// Создание события php artisan make:event UserRegistered
```

```
// Создание слушателя
php artisan make:listener SendWelcomeEmail --event=UserRegistered
// Регистрация слушателя в EventServiceProvider
```

protected \$listen = [
 'App\Events\UserRegistered' => [
 'App\Listeners\SendWelcomeEmail',
],

];

Omber 23

Аутентификация пользователей в Laravel

Аутентификация пользователей в Laravel может быть реализована с помощью встроенных механизмов.

Основные шаги:

- 1. Установка пакета аутентификации: composer require laravel/ui
- 2. Генерация аутентификационных маршрутов и представлений: php artisan ui vue --auth
- 3. Настройка базы данных и миграций.
- 4. Регистрация маршрутов аутентификации в web.php.

ambern 24

Пагинация в Laravel

Пагинация позволяет разбивать большие наборы данных на страницы.

Пример использования пагинации в Eloquent:

```
// Получение данных с пагинацией $users = User::paginate(15);
```

```
// Отображение данных в представлении return view('users.index', compact('users'));
```

```
// В представлении 
{{ $users->links() }}
```

