**Controllers**: A controllerek közvetítik a kapcsolatot a modell (adatelérési réteg) és a nézet (felhasználói felület) között. A HTTP kérések alapján döntenek, hogy milyen adatokat kell lekérni vagy módosítani a modelleken, majd az eredményt továbbítják a megfelelő nézetek felé.

A Laravel controllerek kulcsfontosságú elemei a keretrendszer MVC (Model-View-Controller) struktúrájának. Ezek az osztályok felelősek az üzleti logika és az adatmanipuláció kezeléséért a webalkalmazásokban.

**Middleware-ek**: A Laravel lehetőséget biztosít middleware-ek használatára a controllerekben, amelyek segítenek a kérések előzetes feldolgozásában vagy a kimeneti adatok manipulálásában. Projectünkben a Laravel előre definiált middleware-jein kívül az Sanctum Authentication middleware-t használtuk a felhasználói azonosításhoz, és egy saját EventLog middleware-t az események naplózásához a táblázatban megjelölt útvonalakon.

**Útvonalak kezelése**: A Laravelben a controllerekhez útvonalakat rendelünk, hogy az alkalmazásnak megfelelően irányítsák a kéréseket. Ez lehetővé teszi az alkalmazás logikai szétválasztását és az útvonalak egyszerű kezelését.

Az útvonalak lehetnek dinamikusak, és paramétereket fogadhatnak. Ezek a paraméterek továbbítják az adatokat a kontroller metódusoknak, lehetővé téve az alkalmazás számára, hogy az adott kéréshez kapcsolódó dinamikus adatokat kezelje.

Az útvonalakat a routes/web.php vagy routes/api.php fájlokban definiáljuk. Ezek a fájlok tartalmazzák az alkalmazásunkhoz tartozó útvonalakat, amelyek az URL-eket a megfelelő controllerekhez rendelik.

Táblázat: Controller-ek és Route-ok

|  |  |  |  |
| --- | --- | --- | --- |
| **routes/web.php** | **middleware(s)** | **HTTP metódusok** | **Controller metódusok** |
| **StripeController** |  |  |  |
| /success |  | GET | success |
| /cancel |  | GET | cancel |
| /webhook |  | POST | webhook |

|  |  |  |  |
| --- | --- | --- | --- |
| **routes/api.php** | **middleware(s)** | **HTTP metódusok** | **Controller metódusok** |
| **AuthController** |  |  |  |
| /register | eventlogger | POST | register |
| /login | eventlogger | POST | login |
| /logout | auth:sanctum, eventlogger | POST | logout |
| **ProductController** |  |  |  |
| /product/userFavorites | auth:sanctum | GET | userFavorites |
| /product |  | GET | index |
| /product/{id} |  | GET | show |
| /product/{id}/addToFavorites | auth:sanctum | POST | addToFavorites |
| /product/{id}/removeFromFavorites | auth:sanctum | POST | removeFromFavorites |
| **PurchaseController** |  |  |  |
| /purchase | auth:sanctum | GET | index |
| /purchase/{id} | auth:sanctum | GET | show |
| /purchase/{id} | auth:sanctum, eventlogger | PUT | update |
| /purchase | eventlogger | POST | store |
| **ReservationController** |  |  |  |
| /reservation/checkAvailableDesk |  | GET | checkAvailableDesk |
| /reservation | auth:sanctum | GET | index |
| /reservation/{id} | auth:sanctum | GET | show |
| /reservation/{id} | auth:sanctum, eventlogger | PUT | update |
| /reservation/{id} | auth:sanctum, eventlogger | DELETE | destroy |
| /reservation | eventlogger | POST | store |
| **UserController** |  |  |  |
| /user | auth:sanctum | GET | index |
| /user/{id} | auth:sanctum | GET | show |
| /user | auth:sanctum, eventlogger | PUT | update |
| user/{id} | auth:sanctum, eventlogger | DELETE | destroy |
| /user | auth:sanctum, eventlogger | POST | store |
| **AllergenController** |  |  |  |
| /allergen |  | GET | index |
| **CategoryController** |  |  |  |
| /category |  | GET | index |
| **IngredientController** |  |  |  |
| /ingredient |  | GET | index |
| **DeskController** |  |  |  |
| /desk |  | GET | index |
| **StripeController** |  |  |  |
| / |  | GET | index |
| /checkout |  | POST | checkout |

Model-ek:

**Adatelérési réteg**: A Laravel modellek az adatelérési réteget képviselik a keretrendszerben. Ezek az osztályok kapcsolódnak az adatbázishoz, és lehetővé teszik az adatok lekérdezését, frissítését, létrehozását és törlését az alkalmazásban.

**ORM használata**: A modellek az ORM (Object-Relational Mapping) segítségével működnek, ami lehetővé teszi az objektumok és az adatbázis táblák közötti egyszerű átjárást és műveleteket. Ennek köszönhetően a fejlesztőknek nem kell közvetlenül SQL lekérdezéseket írniuk az adatbázis kezeléséhez.

**Egyszerű használat**: A Laravel modellek könnyen használhatóak és testreszabhatóak. Az alapértelmezett konvencióknak megfelelően a modellek gyakran automatikusan értelmezik az adatbázistáblák struktúráját, így a fejlesztőknek nem kell külön definiálniuk a táblák és az osztályok közötti leképezéseket.

**Kapcsolatok kezelése**: A modellek lehetővé teszik a kapcsolatok kezelését az adatbázisban, például az egy-egyhez, egy-többhöz és sok-többhöz kapcsolatokat. Ez lehetővé teszi az adatok összetett struktúráinak kezelését és manipulációját.

**Validáció és események**: A modellek lehetőséget biztosítanak a beépített validációra, amely lehetővé teszi az adatok ellenőrzését, mielőtt azokat mentenénk az adatbázisba. Emellett eseményeket is kiválthatnak különböző műveletek előtt vagy után, például a mentés vagy a törlés előtt.

**Tesztelhetőség**: A modellek könnyen tesztelhetők, mivel az alapvető adatelérési műveletek egyszerűen szimulálhatók a tesztekben. Ez lehetővé teszi a fejlesztőknek az alkalmazás adatelérési rétegének hatékony tesztelését.

|  |  |  |
| --- | --- | --- |
| **Model név** | **kapcsolat** | **kapcsolt model** |
| Allergen | belongsToMany | ingredients |
| Category | hasMany | products |
| Desk | hasMany | reservation |
|  | hasMany | purchase |
| EventLog | belongsTo | user |
| Image | hasOne | product |
| Ingredient | belongsToMany | allergens |
|  | belongsToMany | products |
| Product | belongsTo | category |
|  | belongsTo | image |
|  | belongsToMany | ingredients |
|  | belongsToMany | purchases |
|  | belongsToMany | users |
| Purchase | belongsTo | desk |
|  | belongsTo | user |
|  | belongsToMany | products |
| Reservation | belongsTo | desk |
|  | belongsTo | user |
| User | hasMany | purchases |
|  | hasMany | reservations |
|  | hasMany | event\_log |
|  | belongsToMany | products |