ASE230 Project 2 – Laravel REST API + Docker + Hugo

Overview

- Re-implemented the Project 1 dealership API using Laravel 12 + Sanctum.
- Eloquent-only data layer; no raw SQL.
- Docker stack with one-command scripts for app + MySQL.
- Hugo documentation + GitHub Pages auto-deploy via Actions.

Goals & Outcomes

- **API parity** with Project 1 plus Laravel best practices.
- **Auth**: Sanctum bearer tokens for all write operations.
- **Data**: ULID users; cars/sales tables with relationships.
- **Ops**: One-command local runner + Docker setup script.
- **Docs**: Hugo site with screenshots and evidence.
- **CI/CD**: GitHub Actions builds docs and publishes to Pages.

Architecture

- Laravel 12 (PHP 8.4) with Sanctum.
- MySQL 8 (Docker service).
- Eloquent models: `User`, `Car`, `Sale`.
- Routes under `/api/*` with `auth:sanctum` for write paths.
- Seeded admin user: `admin` / `Carlo`.

API Catalogue

- `POST /api/auth/login` — Issue bearer token.
- `GET /api/cars` — List cars.
- `GET /api/cars/{id}` — Get car details.
- `POST /api/cars` — Create car (auth).
- `PUT /api/cars/{id}` — Update car (auth).
- `DELETE /api/cars/{id}` — Delete car (auth).
- `GET /api/sales` — List sales.
- `POST /api/sales` — Record sale (auth).

Auth Flow (Sanctum)

1. Client posts credentials to `/api/auth/login`.
2. Password hash verified; Sanctum token created.
3. Token returned; subsequent writes require `Authorization: Bearer <token>`.
4. `auth:sanctum` middleware protects POST/PUT/DELETE routes.

Data Model

- **users**: ULID id, username, role, password (hashed).
- **cars**: make, model, year, price, status (`available|sold`).
- **sales**: car_id, customer_name, sale_price, timestamps.
- **personal_access_tokens**: Sanctum tokens with ULID morphs.
- Relationships: User has tokens; Car has many Sales; Sale belongs to Car.

Migrations & Seeders

- Migrations create users, cars, sales, Sanctum tokens.
- Seeder:
  - Admin user (`admin` / `Carlo`).
  - Sample cars (Supra, Mustang).
- Idempotent seeding (updateOrCreate / firstOrCreate).

Deployment Scripts

- **Local:** `code/run.sh`
  - composer install, `php artisan key:generate`, migrate --seed, serve on :8000.
- **Docker:** `code/setup.sh`
  - Builds app + MySQL, waits for DB, migrate --seed.
- **Compose:** `code/docker-compose.yml` (app + MySQL), `code/Dockerfile` (PHP 8.4).

Testing Aids

- `code/tests/curl.sh` — full flow: login → CRUD cars → sale → delete.
- `code/tests/tests.html` — prompt-based browser harness for manual checks.

Evidence (Screenshots)

code — -zsh — 171×47

```
carlocalipo@Carlos-MBP code % bash setup.sh
[+] Building 0.8s (17/17) FINISHED
 => [internal] load local bake definitions                                    0.0s
 => => reading from stdin 547B                                                0.0s
 => [internal] load build definition from Dockerfile                          0.0s
 => => transferring dockerfile: 736B                                          0.0s
 => [internal] load metadata for docker.io/library/composer:2                 0.3s
 => [internal] load metadata for docker.io/library/php:8.4-cli                0.3s
 => [internal] load .dockerignore                                             0.0s
 => => transferring context: 2B                                               0.0s
 => [stage-0 1/8] FROM docker.io/library/php:8.4-cli@sha256:036def05378dcfdaf6e5fd4c002e3a1751209a27be1a  0.0s
 => => resolve docker.io/library/php:8.4-cli@sha256:036def05378dcfdaf6e5fd4c002e3a1751209a27be1a3ae2f8e6  0.0s
 => [internal] load build context                                             0.3s
 => => transferring context: 832.39kB                                         0.3s
 => FROM docker.io/library/composer:2@sha256:8f3b7d97cd7436dae41d6af0dd6a53853779040ecac26e65544fc317460  0.0s
 => => resolve docker.io/library/composer:2@sha256:8f3b7d97cd7436dae41d6af0dd6a53853779040ecac26e65544fc  0.0s
 => CACHED [stage-0 2/8] RUN apt-get update   && apt-get install -y git unzip libzip-dev libicu-dev libo  0.0s
 => CACHED [stage-0 3/8] COPY --from=composer:2 /usr/bin/composer /usr/bin/composer                       0.0s
 => CACHED [stage-0 4/8] WORKDIR /var/www/html                                                            0.0s
 => CACHED [stage-0 5/8] COPY backend/composer.json ./                                                    0.0s
 => CACHED [stage-0 6/8] RUN composer install --no-dev --no-interaction --prefer-dist --no-progress || t  0.0s
 => CACHED [stage-0 7/8] COPY backend ./                                                                  0.0s
 => CACHED [stage-0 8/8] RUN cp .env.example .env   && composer install --no-dev --no-interaction --pref  0.0s
 => exporting to image                                                        0.0s
 => => exporting layers                                                       0.0s
```

Laravel Dealership API Tester

Login

Login as admin

Cars

List Cars | Create Car (secured) | Update Car (secured) | Delete Car (secured)


```
[
  {
    "id": 6,
    "make": "Lamborghini",
    "model": "Aventador SVJ",
    "year": 2020,
    "price": "785000.00",
    "status": "sold",
    "created_at": "2025-12-01T21:02:31.000000Z",
    "updated_at": "2025-12-01T21:02:46.000000Z"
  },
  {
    "id": 1,
    "make": "Toyota",
```

tests — -zsh — 138×47

```
carlocalipo@Carlos-MBP tests % bash curl.sh
AUTH RAW: {"token":"5|hpN6BWLsxQxZjXs3yRqKNpehIafpWiyad1oko9MB16e22b15","user":{"id":"01kbdv5ry1436mqx8ckke6tvea","username":"admin","role
":"admin"}}
TOKEN=5|hpN6BWLsxQxZjXs3yRqKNpehIafpWiyad1oko9MB16e22b15
[
  {
    "id": 6,
    "make": "Lamborghini",
    "model": "Aventador SVJ",
    "year": 2020,
    "price": "785000.00",
    "status": "sold",
    "created_at": "2025-12-01T21:02:31.000000Z",
    "updated_at": "2025-12-01T21:02:46.000000Z"
  },
  {
    "id": 1,
    "make": "Toyota",
    "model": "Supra",
    "year": 2021,
    "price": "56000.00",
    "status": "sold",
    "created_at": "2025-12-01T20:55:25.000000Z",
    "updated_at": "2025-12-01T21:03:37.000000Z"
  }
]
{
  "id": 7
}
CAR_ID=7
[
  {
    "id": 7,
    "make": "Toyota",
    "model": "Supra",
    "year": 2021,
    "price": "55000.00",
```

```
[carlocalipo@Carlos-MBP code % bash run.sh
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Nothing to install, update or remove
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

   INFO  Discovering packages.

  laravel/pail ..................................................... DONE
  laravel/sail ..................................................... DONE
  laravel/sanctum .................................................. DONE
  laravel/tinker ................................................... DONE
  nesbot/carbon .................................................... DONE
  nunomaduro/collision ............................................. DONE
  nunomaduro/termwind .............................................. DONE

81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

   INFO  Application key set successfully.


  Dropping all tables ..................................................... 63.69ms DONE

   INFO  Preparing database.

  Creating migration table ................................................. 12.90ms DONE

   INFO  Running migrations.

  0001_01_01_000000_create_users_table ..................................... 27.93ms DONE
  0001_01_01_000001_create_cache_table ..................................... 10.39ms DONE
  0001_01_01_000002_create_jobs_table ...................................... 25.28ms DONE
  2025_02_15_000100_create_cars_table ....................................... 9.96ms DONE
  2025_02_15_000200_create_sales_table ..................................... 23.48ms DONE
  2025_12_01_195508_create_personal_access_tokens_table .................... 25.73ms DONE
```

Configuration Notes

- `.env` set for MySQL host `db` (Docker) or `127.0.0.1` locally.
- Sanctum guard added; ULID morph for tokens to avoid truncation.
- CORS enabled for `/api/*`.

Lessons Learned

- Composer lock demanded PHP 8.4 in Docker to match deps.
- Sanctum + ULID requires `ulidMorphs` on `personal_access_tokens`.
- Canonical URLs for GitHub Pages subpath to fix link issues.
- Seeders should be idempotent to avoid duplicate admin errors.

Troubleshooting

- **Auth 500 + tokenable_id truncation**: use `ulidMorphs` in tokens migration.
- **Pages links doubled path**: set `baseURL` and use canonical URLs; adjust menu links.
- **Docker compose DB not found**: ensure services running, rerun `setup.sh`.

Detailed Endpoint Behaviors

- Cars POST/PUT validate make/model/year/price; status optional on update.
- Sales POST validates car exists; marks car as `sold` after sale creation.
- Deletes return `{deleted: true}` and 404 when re-fetching.

Security Considerations

- Sanctum tokens only; no sessions/cookies.
- Bearer required for all write operations.
- Validation on inputs; 404/401/403 responses for bad states.

Performance & Stability

- Simple query patterns via Eloquent.
- Seeded data for instant testing.
- Docker isolates app + DB; consistent local/dev environment.

Documentation & Pages

- Hugo site under `presentation/` with custom layouts (no external theme).
- Docs for Project1/Project2, Portfolio, screenshots, links to tests.
- GitHub Actions (`.github/workflows/gh-pages.yml`) builds Hugo and deploys to Pages.

How to Run (Local)

```
cd code
./run.sh
```

Hit `http://localhost:8000/api` and use `curl.sh` or `tests.html`.

How to Run (Docker)

```
cd code
./setup.sh
```

App on `http://localhost:8000/api`. Credentials: `admin` / `Carlo`.

Testing Flow (Sample)

1. Login → token
2. Create car (auth)
3. Update price (auth)
4. Record sale (auth)
5. List sales
6. Delete car (auth) → 404 on fetch

Evidence Narrative

- **setup.sh**: shows successful build, migrate, seed with app+db containers.
- **Browser tester**: manual buttons for auth/cars/sales returning JSON.
- **cURL flow**: scripted end-to-end outputs confirming CRUD + sale.

Future Enhancements

- Add pagination and filters to `/api/cars`.
- Add OpenAPI/Swagger documentation.
- Add PHPUnit feature tests for controllers.
- Add CI to run tests on push.

Key Files Reference

- Routes: `code/backend/routes/api.php`
- Controllers: `AuthController`, `CarController`, `SaleController`
- Models: `User` (ULID, HasApiTokens), `Car`, `Sale`
- Migrations: users, cars, sales, personal_access_tokens (ULID morph)
- Seeders: `DatabaseSeeder` (admin + sample cars)
- Scripts: `run.sh`, `setup.sh`
- Tests: `code/tests/curl.sh`, `code/tests/tests.html`

Summary

Project 2 delivers a Laravel-based dealership API with Sanctum auth, Dockerized deployment, one-command scripts, comprehensive tests, and a Hugo site published via GitHub Pages, fully re-implementing Project 1 with modern tooling and CI/CD.