Instrukcja Obsługi.

By skorzystać z Ollegro lokalnie będą nam potrzebne 2 rzeczy. Zainstalowana nowa wersja Node >= 18.0.0 oraz Docker. Po ich zainstalowaniu trzeba pobrać moduły wykonujemy to poprzez dokonanie w folderach ./packages/api oraz ./packages/client komendy „npm install –prefix ./”. Następnie w ./packages/api trzeba umieścić odpowiednie klucze prywatne w plikach .env oraz .env.sample . Gdy wykonaliśmy te czynności wystarczy że w głównym folderze projektu odpalimy następująco 3 komendy równolegle.

1. Docker-compose up - odpali nam ona instancje redisa w kontenerze
2. Npm run start-server - włączy instancje serwera
3. Npm run start-client - włączy instancje clienta

By odpalić testy:  
W głównym folderze wpisujemy:

1. Docker-compose up - testy nadal korzystaja z instacji redisa

W folderze ./packages/api

1. Npm run test

Biblioteki:

1. TRPC – nakładka na serwer pozwalająca tworzyć szybko endpointy wraz z end-to-end typesafety.
2. Prisma – biblioteka pomagająca deweloperom zczytywać i pisać dane do bazy danych w intuicyjny sposób przy zachowaniu typesafety.
3. Firebase – biblioteka pozwalająca na połączenie się z ich bazą danych ( w naszym przypadku do przechowywania zdjęć )
4. Redis – cache naszego serwera, potrzebna jest ona by sprawdzać czy sesje użytkowników są nadal aktualne.
5. Jest – bliblioteka do testowania ponieważ używa ona również typescripta, ułatwia to o wiele testowanie Trpc.
6. Bcrypt – hashuje ona hasła, by mogły być przechowywane bezpiecznie.
7. ReactQuery – Trpc Client pod spodem operuje tak naprawdę na niej by cachować requesty.
8. NodeMailer – umożliwia ona nam połączenie się z pocztą, by wysyłać z niej maile.
9. Zod – pozwala na sprawdzanie typów danych w czasie działania serwera.
10. Cors – Ponieważ client i serwer działają na różnych portach, defaultowo mają one zakaz komunikowania się pomiędzy sobą. Cors zezwala na komunikacje względem wybranych portów, witryn.

Wzorzec Projektowy wykorzystany głównie w projekcie to Singleton. Wykorzystany on został do utworzenia wszelkiego rodzaju instancji baz danych czy chociażby redisa. Trpc również w swój sposób jest adapterem, ponieważ umożliwia zrozumienie serwerowych end-pointów na kliencie.

Inwencja własna:

1. Otypowanie end-to-end całej aplikacji, co pozwala na łapanie błędów podczas kompilacji.
2. Użycie JWT podczas autentykacji użytkownika, co pozwala na bezpieczniejsze logowanie.
3. Użycie oddzielnej bazy danych do przechowywania zdjęć (skalowalność)