I using of design pattern includes:

Strategey :

for develop system if in future we will want add feature it possible and switch between that's and also services To be separated

Repository :

for develop datasources we using of this pattern for example : if in future we will want change db to postgres it possible

I using of Middleware for check user is authed or no then show him result for search book

I using of test case for :

login

register

refresh token

get list book

I using of Docker and Docker Compose for project

Redis Cache with important method Cache::remember() :

I using of Cache::remember() because this method will cache mysql query for 10 min And if item wasn't in redis cache first will add to redis And then items cached show to user and is optimize and short code

EndPoints are :

Search book:

Search book with keyword method post

header : Authorization Bearer eyJ0eXAiOiJK. . . . .

input : keyword

method : post

http://127.0.0.1:8000/api/search/book

Login:

Login with email and password and get jwt token input : email,password method : post

http://127.0.0.1:8000/api/auth/login

Register

Register user with name,email,password

input : name,email,password

method : post http://127.0.0.1:8000/api/register

Refresh token

Refresh token

method : post

http://127.0.0.1:8000/api/register

Logout token

Logout token

method : post

http://127.0.0.1:8000/api/auth/logout