Saba moradi

A fisrt project about book store

Package	Version
alembic	1.14.1
amqp	5.3.1
annotated-types	0.7.0
antiorm	1.2.1
anyio	4.8.0
app	0.0.1
asgiref	3.8.1
billiard	4.2.1
Booktype	1.5
celery	5.4.0
cffi	1.17.1
click	8.1.8
click-didyoumean	0.3.1
click-plugins	1.1.1
click-repl	0.3.0
cryptography	44.0.0
databases	0.9.0
db	0.1.1
distlib	0.3.9
Django	5.1.5
ecdsa	0.19.0

FASTAPI   0.115.7

FILELOCK   3.17.0

FUNCTIONS   0.7.0

GREENLET   3.1.1

H11   0.14.0

HTTPLIB2   0.22.0

IDNA   3.10

INICONFIG   2.0.0

JWT   1.3.1

KOMBU   5.4.2

Mako   1.3.8

MarkupSafe   3.0.2

OAUTH   1.0.1

OAUTH2   1.9.0.POST1

PACKAGING   24.2

PASSLIB   1.7.4

PIP   24.2

PLATFORMDIRS   4.3.6

PLUGGY   1.5.0

PROMPT_TOOLKIT 3.0.50

PSYCOPG2   2.9.10

PSYCOPG2-BINARY 2.9.10

PYASN1   0.6.1

PYCPARSER   2.22

PYDANTIC   2.10.6

PYDANTIC_CORE 2.27.2

PYPARSING   3.2.1

PYTEST   8.3.4

PYTHON-DATEUTIL   2.9.0.POST0

PYTHON-JOSE   3.3.0

REDIS   5.2.1

RSA   4.9

SCHEMAS   0.7.1

SETUPTOOLS   75.8.0

SIMPLEJSON   3.19.3

SIX   1.17.0

SNIFFIO   1.3.1

SQLALCHEMY   2.0.37

SQLPARSE   0.5.3

STARLETTE   0.45.3

TYPING_EXTENSIONS 4.12.2

TZDATA   2025.1

UVICORN   0.34.0

VINE   5.1.0

VIRTUALENV   20.29.1

WCWIDTH   0.2.13

DOD

IN THIS PROJECT WE TRY TO MAKE A LIBRARY BOOK SYSTEM.

IN THIS SYSTEM WE MAKE A DATABASE AND MAKE A CRUD WITH TABLE.

WE NEED A AUTH SERVICE AND SOME SUB SERVICE.

IN THIS PROJECT I USE SOME PACKAGE (FASTAPI , CELERY, SQLALCHEMY , PSYCPG2 , PYDANTIC ,...)

FOR RUN A API SQLALCHEMY FOR CONNECTING A DATABASEFASTAPI A PACKAGE AND FRAMEWORK FOR BUILDING API SO QUICKLY ALSO I USE PostgreSQL FOR CREATE A DATABASE AND CONNCTED TO PyCharm

AFTER THAT I WRITE A MODELS FOR CONNECT TO DATABASE AND HOLDING DATA AND MIGRATION THAT WITH ALEMBIC(ALEMBIC USEFUL FOR FASTAPI)

AFTER THAT WITH ROTHERAPI AND FAST API BUILD SERVICE COMDITION FOR TABLES AND CRUDS Pydantic IS VERY SIMPLE AND POWERFUL

.ITS USEFUL FOR MANAGE DATABASES AND CREATE API.

WHEN WE USE A PAYDANTIC , WE SHOULD CREATE A MODELS THAT Inheritance BaseModel WE USE JWT , PASSLIB .. FOR AUTHETICATION, SIGN IN SIGN UP AND I USE CELERY AND REDIS FOR MAKE QUEUE FOR RESERVE BOOK WHEN ANYBODY DONT RESERVED ANY BOOKS.....

CHECKLIST OF DOD

- Make required database table data model and relations between them and put sample (at least 10 rows of data) in them
  - User
  - Author
  - Customer
  - Book
  - City
  - Genre
  - Reservation
- Make required CRUD service for:
  - Book
  - User
  - Customer
  - Reservation
- Make required general service (that usually is something that is using some smaller service)
  - User auth - sign up, sign in, permission management functionalities
  - Reservation service - reservation checks and functionalities
  - purchasing service

- Make required apis for all crud services and general services
  - Required method to implement:
    - Create → POST
    - Read → GET
    - Update → PUT
    - Delete → DELETE
  - Required services
    - Simple cruds
      - Book
      - User
      - Customer
      - Reservation
    - General service
      - Auth
      - Reservation
      - purchasing
- General business logic consideration
  - Users login with password and then OTP, instead of using an sms service, write an sample abstract version of sms service that instead of sending sms, print out OTP code in the console.
  - A place for purchase plus (monthly 50,000 Toman) and premium (monthly 200,000 Toman) membership
  - Different membership status have different permissions
    - Free users :
      - Cannot reserve books at all
      - Just can fetch book data
    - Plus users can do anything in free plus :
      - Reserve books for at most 1 week (7 consecutive days)
      - Pay 1000 Toman for each reservation day
      - If read more than 3 different books in previous month (30 days) get a 30% discount
      - If paid more than 300,000 Toman in previous two months (60 days) get a 100% discount (free reservation)
      - Can reserve up to 5 different book units simultaneously
    - Premium users can do anything in plus and :
      - Can reserve a book for at most 2 weeks (14 consecutive days)
      - In reservation queue for a book , they are prioritized

- Can reserve up to 5 more different book units (a total of 10 different book units) simultaneously
- Reservation
  - If there are more than 0 remaining units for a book to reserve, an instant reserve is issued on reserve request. Meaning another unit of book considered reserved, and the reservation time of that order starts at the moment of order placement
  - The number of books that instant reserved should not exeed the number of units of that book (if we have 3 unit for a book, we cannot instant reserve it 4 times), in other words, remaining unit for a book should not get negative
  - If there are 0 remaining units for a book to reserve, a scheduled reserve is issued on reserve request. Meaning the requested customer goes into a queue with custom ordering between customers and any time a unit of book released, we remove first customer from that queue and issue an instant reserve of that book for that customer issued on the time of book unit release
  - The order of customers in reservation queue defined in this way
    - Between all users, the ones with premium subscription model on the time of unit release have a higher priority and moves to the beginning of the queue before any other plus customer
    - Between all users with same subscription models, the one who scheduled a reserve sooner have a higher priority
  - Users can request to exit the queue any time they want
  - Users cannot instant reserve more than number of their remaining number of reservation limit. If the instant reserve request was from user it returns error. If it was from the queue, fails silently, user is removed from queue and not reservation issue for that user, we try to instant reserve that book for next ones in queue
  - If the current amount of wallet of user that issue a reserve, either instant or scheduled, is not enough for their order, return error, redirect to purchase api for the remaining required amount to charge their account.
  - On instant reserve the required amount of money is subtracted from their wallet. If on issuing instant reserve from queue the customer, the customer doesn't have sufficient amount of charge, the request fails silently, user is removed from queue and not reservation issue for that user, we try to instant reserve that book for next ones in queue
- Purchase
  - A simple place for people to charge their wallet

- ~~A place to request for upgrade / renew their subscription model. Remove the required amount from their wallet and change their subscription model and / or deadline~~
- Other cruds
  - Each user - customer - author 's data can only be changed by him/herself
  - Each book's data can only be changed by its author(s)
  - Reservation data cannot directly changed. It can only changed through reservation service
- Auth
  - Implement handling auth, using jwt. This token used for auth process of users in any place. Only place that can be available without auth process is viewing book data and user sign up
  - Whole auth processes should implement as a service and get called from requested places
- Admin
  - Admins can revoke any users token except admins
  - Admins can end reservations for a user before its end time
  - Admins can view a books current holders and scheduled reserver and thier orders and remove any one
- General technical consideration
  - All APIs have validation based on their data
  - Errors status codes handled currently based on their messages, reason and general guidelines for http errors status codes
  - You write services separated from apis. Apis are just something that calls those services.
  - Each data should be an entity object in system. These entities should validate themselves. Entity properties should not manipulated directly and can only be manipulated using its method