Tables and ORM objects:

TABLES:

Users:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column name | Type | Nullable | Const | Unique | Note |
| id | Serial(integer) | False | True | True |  |
| name | String(50) | False | False | True |  |
| password | String(160) | False | False | False | Save the hash, with salt |
| mail | String(255) | False | True | True |  |
| token |  |  |  | True |  |

TODOs:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column name | Type | Nullable | Const | Unique | Note |
| id | Serial(integer) | False | True | True |  |
| Title | Text | False | False | False |  |
| Description | Text | True | False | False |  |
| Priority | Enum(‘low’, ‘medium’, ‘high’) | False | False | False | some kind of enum (ready, in progress, paused, blocked, done) |
| Deadline | Date | False | False | False | Timestamp |
| Creator | integer | False | True | False | the user who created the todo (user id) |
| Status | Enum(‘ready’, ‘in progress’, ‘paused’, ‘blocked’, ‘done’) | False | False | False | some kind of enum (pending, in progress, tests, finished) |
| Reminder | Timestamp with time zone | True | False | False | maybe date or week before (cron) |
| Tags | Text[] | True | False | False | List of strings |

User TODO relation:

key



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column name | Type | Nullable | Const | Unique | Note |
| User id | integer | False | True | false | Foreign key users(id) |
| Todo id | integer | False | True | False | Foreign key todos(id) |

User todo relation

User\_id

Todo\_id

Todos

Id

Title

description

priority

deadline

creator

status

reminder

tags

Users

Id

Name

Password

mail

Todo:

Attributes:

1. id
2. Title – title
3. Description - description
4. Priority – some kind of enum
5. Deadline – time and date
6. Creator – the user who created the todo
7. Status – some kind of enum (pending, in progress, tests, finished)
8. Tags ?
9. Reminder – maybe date or week before

User:

Attributes:

1. Id – Const, not null
2. User name – not null
3. Password – not null
4. Mail – not null
5. Token?

API:

1. Post:
   1. Create user
   2. Create a todo ( will create tag, if needed)
2. PUT:
   1. Update user information ( only user name, password, mail)
   2. Update todo information ( only Title, description, priority, deadline, status, reminder, tags)
3. GET:
   1. Get user information
   2. Get todo’s information
   3. Get todos by tag
   4. Get todos by user
   5. Get todos by creator
   6. Get users by todo
4. DELETE:
   1. Delete user
   2. Delete todo

|  |  |  |
| --- | --- | --- |
| Request type | url | rule |
| Get | /api/v1/user | Get user information |
| Get | /api/v1/todo | Get todo’s information |
| Get | /api/v1/todo\_by\_tag | Get todos by tag |
| Get | /api/v1/todo\_by\_user | Get todos by user |
| Get | /api/v1/todo\_by\_creator | Get todos by creator |
| Get | /api/v1/user\_by\_todo | Get users by todo |
| Post | /api/v1/user | Create user |
| Post | /api/v1/todo | Create a todo |
| Put | /api/v1/user | Update user information |
| Put | /api/v1/todo | Update todo information |
| Delete | /api/v1/user | Delete user |
| Delete | /api/v1/todo | Delete todo |

Scheme

DB pod

Beackend pod

Frontend pod

frotend server – ngnix

Actual app - react

DB - postgreSQL

Server – fastapi – python

Will send mail to user