Tables and ORM objects:

TABLES:

Users:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Nullable | Const | Unique | Note |
| id | False | True | True |  |
| name | False | False | True |  |
| password | False | False | False | Save the hash, with salt |
| mail | False | True | True |  |
| token | ¯\\_(ツ)\_/¯ | ¯\\_(ツ)\_/¯ | True | ¯\\_(ツ)\_/¯ |

TODOs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Nullable | Const | Unique | Note |
| id | False | True | True |  |
| Title | False | False | False |  |
| Description | True | False | False |  |
| Priority | False | False | False | some kind of enum |
| Deadline | False | False | False | Timestamp |
| Creator | False | True | False | the user who created the todo |
| Status | False | False | False | some kind of enum (pending, in progress, tests, finished) |
| Reminder | True | False | False | maybe date or week before (cron) |
| Tags | True | False | False | List of strings |

User TODO relation:

key



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Nullable | Const | Unique | Note |
| User id | False | True | false |  |
| Todo id | False | True | False |  |

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?

Tags:

Attributes:

1. Tag text
2. Todo’s it appear in

Scheme

DB pod

Beackend pod

Frontend pod

frotend server – ngnix

Actual app - react

DB - postgreSQL

Server – fastapi – python

Will send mail to user

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. 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 |
| 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 |