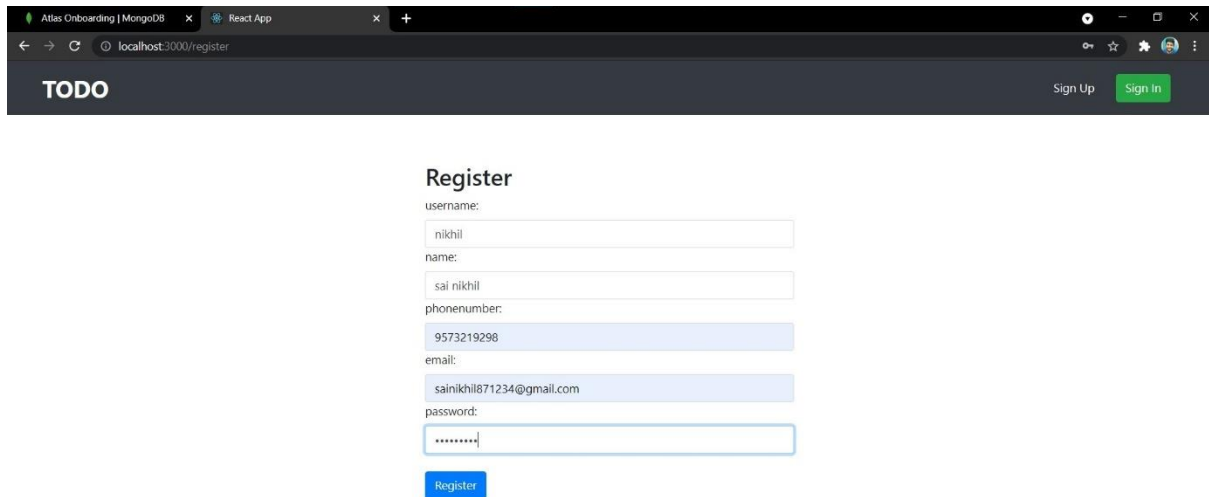


# TO DO APPLICATION

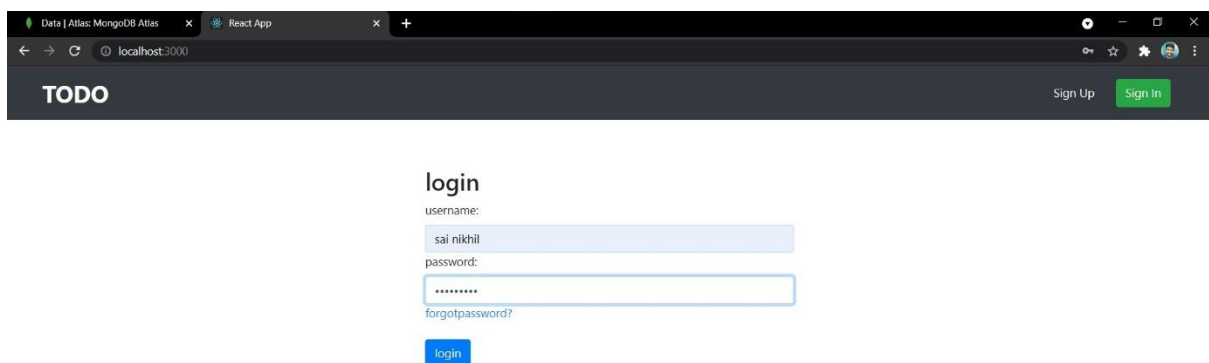
## STEP 1:

User has to Register before going to use our application



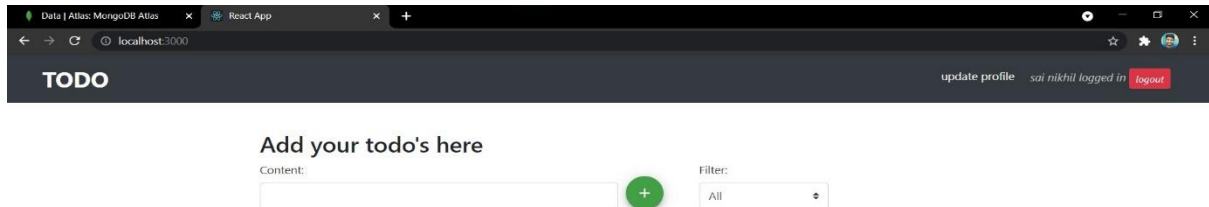
The screenshot shows a web browser window with the URL `localhost:3000/register`. The page has a dark header with the word "TODO" on the left and "Sign Up" and "Sign In" buttons on the right. The main content area is titled "Register" and contains a form with the following fields: "username:" (value: nikhil), "name:" (value: sai nikhil), "phonenumber:" (value: 9573219298), "email:" (value: sainikhil871234@gmail.com), and "password:" (value: masked with asterisks). A blue "Register" button is at the bottom of the form.

## STEP 2: After registration , the user has to login

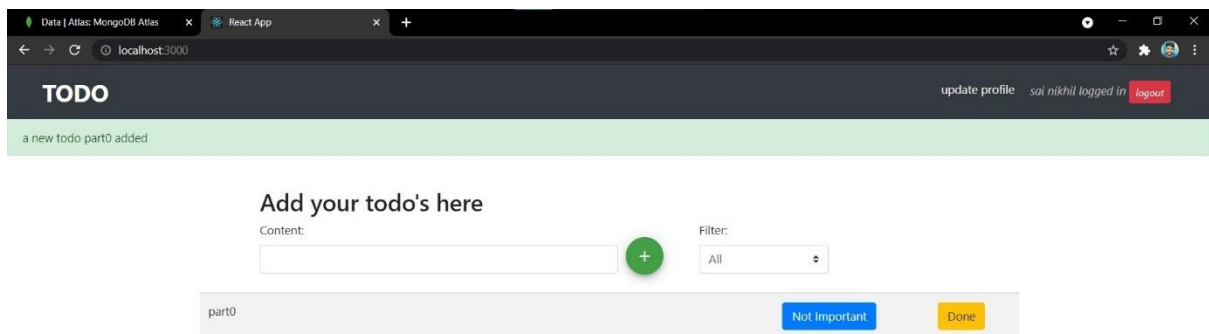


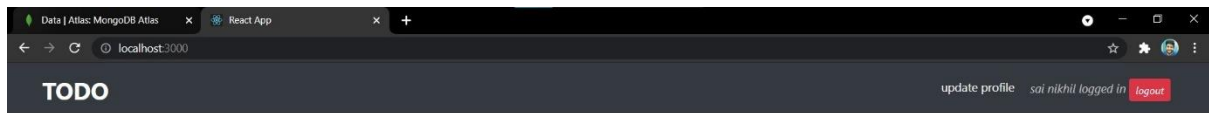
The screenshot shows a web browser window with the URL `localhost:3000`. The page has a dark header with the word "TODO" on the left and "Sign Up" and "Sign In" buttons on the right. The main content area is titled "login" and contains a form with the following fields: "username:" (value: sai nikhil), "password:" (value: masked with asterisks), and a link "forgotpassword?". A blue "login" button is at the bottom of the form.

**STEP 3 :** whenever login has successfully done the Empty To do page appears



**STEP 4 :** Now user can add their tasks





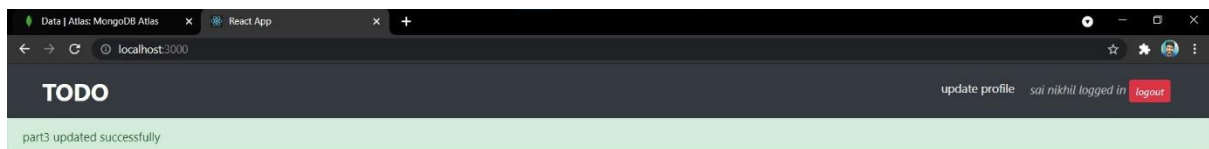
**Add your todo's here**

Content:  + Filter: All

part0	Not Important	Done
part1	Not Important	Done
part2	Not Important	Done
part3	Not Important	Done
part4	Not Important	Done

**STEP 5 :** user can prioritise their specific tasks from

**NOT IMPORTANT → IMPORTANT**

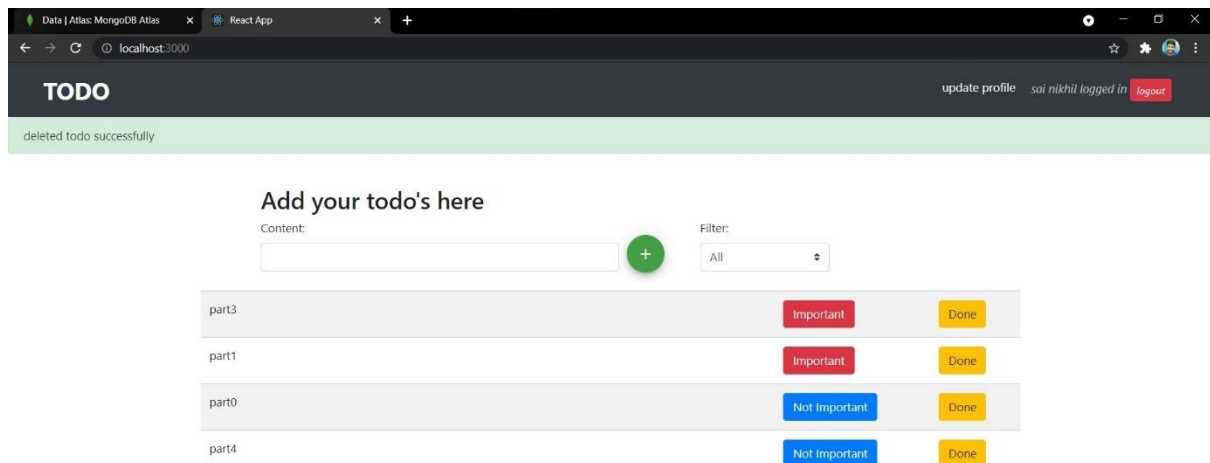
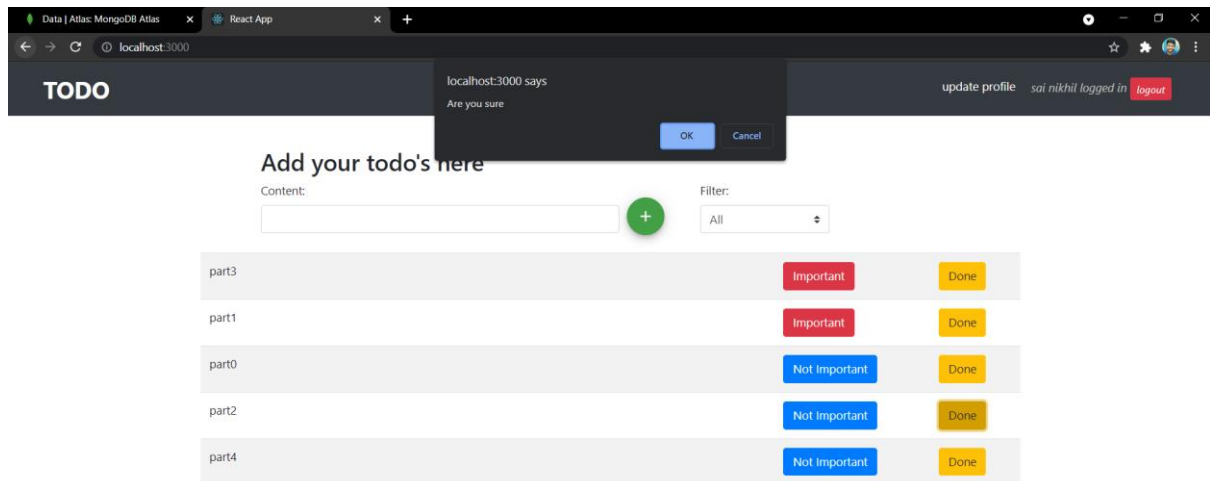


**Add your todo's here**

Content:  + Filter: All

part3	Important	Done
part0	Not Important	Done
part1	Not Important	Done
part2	Not Important	Done
part4	Not Important	Done

**STEP 6 :** whenever the specific task has successfully done they can go for done option  
i.e.; task deleted



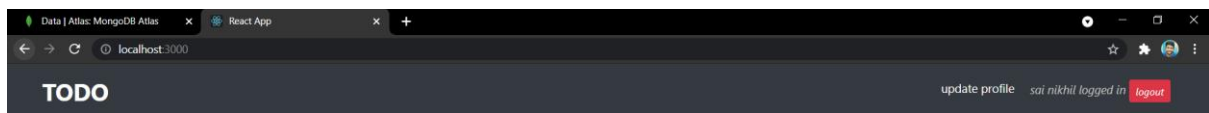
STEP 7 : user can filter their tasks

## IMPORTANT



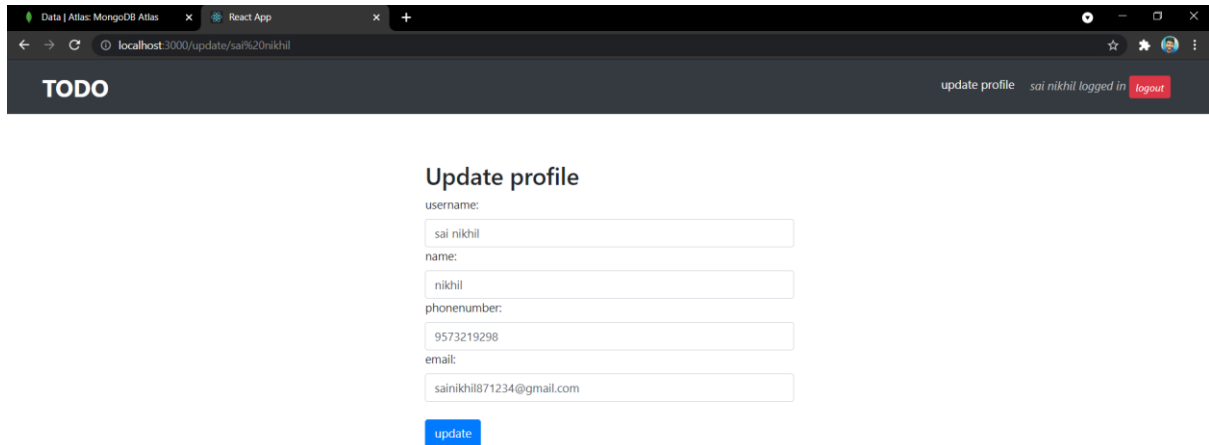
Add your todo's here		
Content:		Filter:
	+	Important
part3	Important	Done
part1	Important	Done

## NOT IMPORTANT



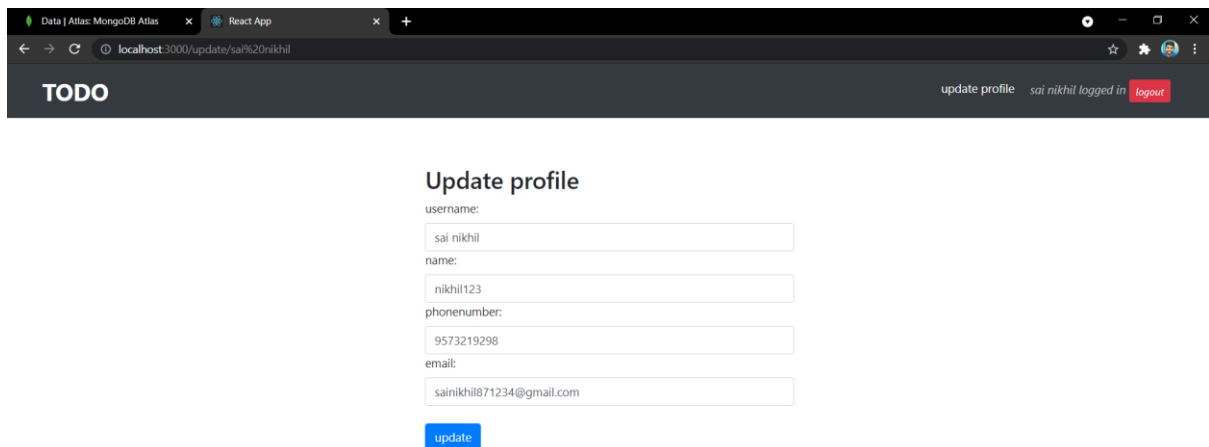
Add your todo's here		
Content:		Filter:
	+	Not Important
part0	Not Important	Done
part2	Not Important	Done
part4	Not Important	Done

## STEP 8 : user can update their profile

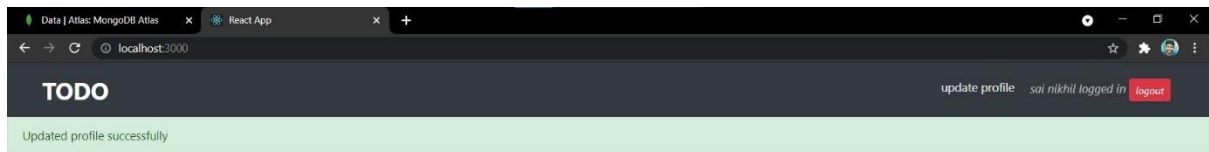


The screenshot shows a web browser window with the address bar displaying `localhost:3000/update/sai%20nikhil`. The browser tabs include 'Data | Atlas: MongoDB Atlas' and 'React App'. The application header is dark grey with the word 'TODO' on the left and 'update profile sai nikhil logged in logout' on the right. The main content area is white and contains the 'Update profile' form. The form has four input fields: 'username' with the value 'sai nikhil', 'name' with the value 'nikhil', 'phonenumber' with the value '9573219298', and 'email' with the value 'sainikhil871234@gmail.com'. A blue 'update' button is located below the email field.

Here I'm trying with name field



This screenshot is similar to the previous one, showing the 'Update profile' form. However, the 'name' field now contains the value 'nikhil123'. All other fields (username, phonenumber, email) and the 'update' button remain the same. The browser window and application header are identical to the previous screenshot.



Add your todo's here

Content:  + Filter: All

part1	Important	Done
part3	Important	Done
part0	Not Important	Done
part4	Not Important	Done

**STEP 9 :** If a user forget their password they can go for **Forget password** option

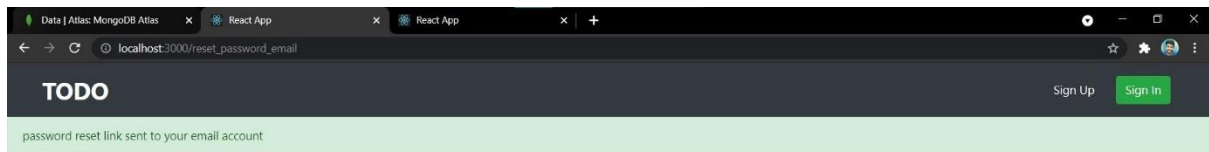


### Reset Password

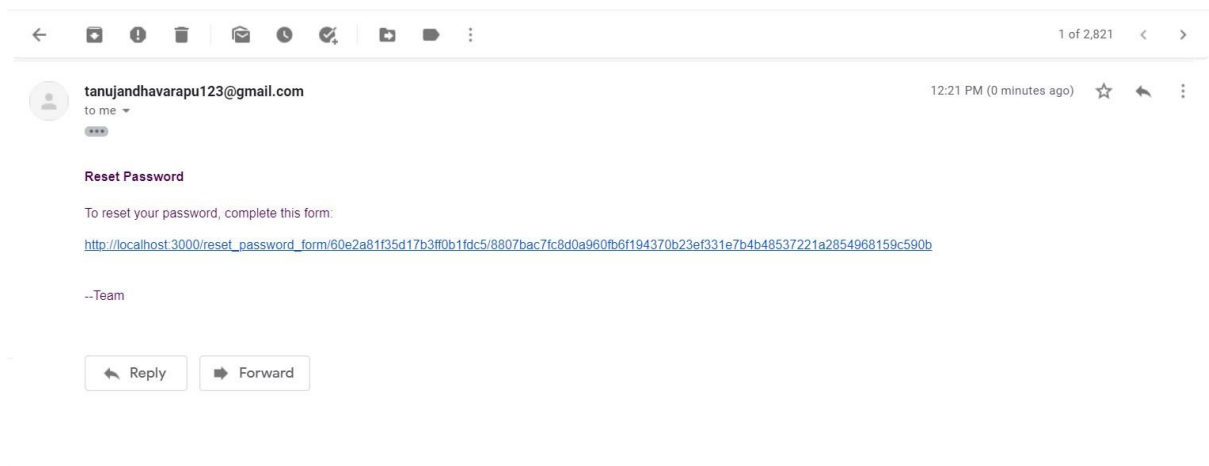
Email:

sainikhil671234@gmail.com

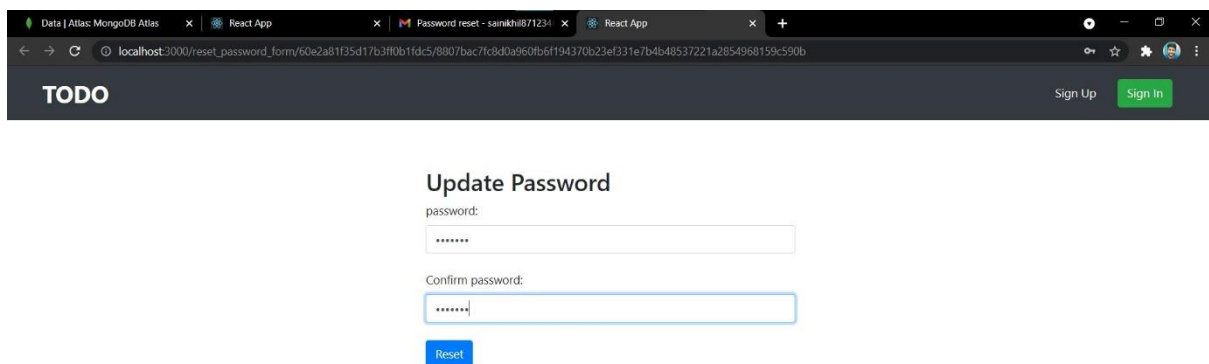
< Submitting...



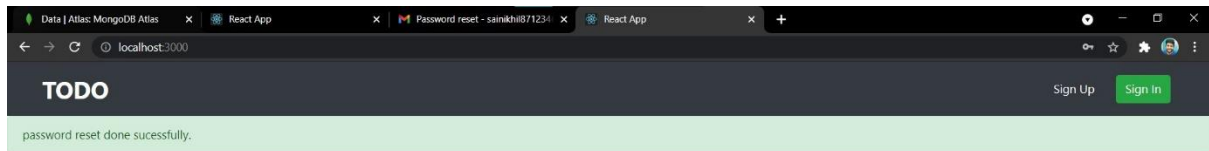
Reset Password  
email sent successfully



After clicking on that link a form will be displayed for resetting the password







## login

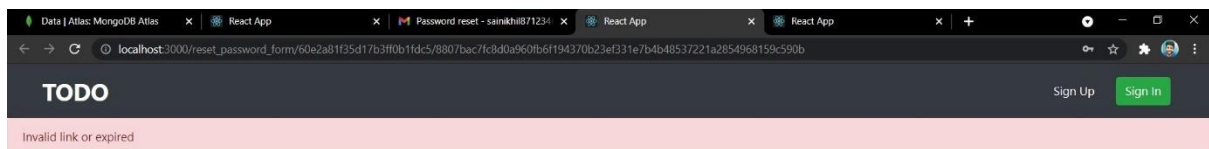
username:

password:

[forgot password?](#)

login

The specific link is only valid for one time , if they go for multiple times it won't work



## Update Password

password:

Confirm password:

Reset

These are the data stored in Mongodb Database

DATABASES: 2 COLLECTIONS: 4

+ Create Database

Q NAMESPACES

▼ todoapddb

todolists

tokens

users

▶ todoapddb-test

Visualize your data

Refresh

todoapddb.users

COLLECTION SIZE: 247B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 60KB

Find

Indexes

Schema Anti-Patterns

Aggregation

Search Indexes

INSERT DOCUMENT

FILTER {"filter": "example"}

Find Reset

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("60e2a81f35d17b3ff0b1fdc5")
phoneNumber: "9573219298"
email: "sainikhil871234@gmail.com"
> todolist: Array
  username: "sai nikhil"
  name: "nikhil123"
  passwordHash: "$2b$10$7MAGEe0bNlYfNPKXHLzkuDORyAtcg7ZoOmvtMyj0II1V/51x6DEC"
  __v: 8
```

FILTER {"filter": "example"}

Find Reset

QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("60e2a91435d17b3ff0b1fdc9")
content: "part0"
important: false
user: ObjectId("60e2a81f35d17b3ff0b1fdc5")
__v: 0
```

```
_id: ObjectId("60e2a92d35d17b3ff0b1fdca")
content: "part1"
important: true
user: ObjectId("60e2a81f35d17b3ff0b1fdc5")
__v: 0
```

```
_id: ObjectId("60e2a93a35d17b3ff0b1fdcc")
content: "part3"
important: true
user: ObjectId("60e2a81f35d17b3ff0b1fdc5")
__v: 0
```