

NEOFI CHAT APP

1. Friends Recommendation: Endpoint: GET /api/suggested-friends/<user_id>

Functionality: Allows to send a GET request to the application with a user_id and it should return the top 5 recommended friends for that user.

This api returns the top 5 friends using Jacard Indexing .

Jaccard Index Calculation: To find the similarity between two users, you calculate the Jaccard Index between their interest sets. The Jaccard Index is defined as:

$$J(A, B) = (|A \cap B|) / (|A \cup B|)$$

Where:

A and B are sets of interests of two users.

$|A \cap B|$ is the size of the intersection of the sets (i.e., the number of common interests).

$|A \cup B|$ is the size of the union of the sets (i.e., the total number of unique interests from both users).

The Jaccard Index ranges from 0 to 1, where 0 means no similarity, and 1 means the sets are identical.

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/api/suggested-friend/933`. The response is a JSON array of 5 objects, each representing a user with their ID, name, age, and a list of interests.

```
[{"id":29,"name":"User 29","age":100,"interests":{"dancing":48,"music":26,"travelling":31,"movies":79,"photography":14}}, {"id":154,"name":"User 154","age":85,"interests":{"music":100,"cars":87,"photography":35,"drawing":45}}, {"id":249,"name":"User 249","age":85,"interests":{"photography":51,"singing":11,"dancing":12,"music":30}}, {"id":313,"name":"User 313","age":99,"interests":{"computers":72,"cars":24,"photography":30,"music":60}}, {"id":339,"name":"User 339","age":91,"interests":{"photography":63,"movies":33,"cars":80,"computers":31,"music":40,"dancing":21}}]
```

1. User Login: Endpoint: POST /api/login/

Functionality: Allows users to log in to their account by providing their credentials (username/email and password).

Output: If the login is successful, return an authentication token or a success message with the user details.

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/api/login/`. The request body is form-data with the following fields:

Key	Value	Description
username	rajojha2	
email	rajojha2@mail.com	
password	lava@1234	

The response is a JSON object with a single key-value pair:

```
1 {  
2   "key": "4851967def8abb38e11c6a1bc54af1580c9e7383"  
3 }
```

2. Get Online Users: Endpoint: GET /api/online-users/

Functionality: Retrieves a list of all online users who are currently available for chat.

Note: Every user comes online when he's logged in. The user can be classified as offline when either he triggers logout himself, or due to the authentication token expiry caused by the user's inaction for the token expiry time.

Output:

Returns a list of online user objects with their details, such as username and status.

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/api/online-users`. The request headers are:

Key	Value	Description
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Token 4851967def8abb38e11c6a1bc54af1...	

The response is a JSON array of user objects:

```
1 [{"username": "ojharaaj"}, {"username": "rajojha1"}, {"username": "rajojha2"}, {"username": "rajojha3"},  
2   {"username": "rajojha4"}]
```

4. Send a Message: Endpoint: WEBSOCKET /api/chat/send/

Functionality: Allows a user to send and receive instant messages to another user who is online.

Output:

If the recipient is online and available, send the message to the recipient in JSON and return a success message or status code.

If the recipient is offline or unavailable, return an error message or status code.

Raw

ws://127.0.0.1:8000/ws/chat/1/

Docs

Feedback

Save

ws://127.0.0.1:8000/ws/chat/1/

Disconnect

Params

Headers

Settings

New message

Save Message

1

{

"message": "Hi,how are you"

}

Text

Send

Messages

CONNECTED

Search

All Messages

Clear Messages

↓

{

"id": 18,

"text": "fine thank you",

"attachment": null,

"timestamp": "2023-09-12T19:56:45.344...

}

01:26:45

↓

↓

{

"id": 17,

"text": "Hi,how are you",

"attachment": null,

"timestamp": "2023-09-12T19:56:16.099...

}

01:26:16

↓

↑

{

"message": "Hi,how are you"

}

01:26:16

↓

✓

Connected to ws://127.0.0.1:8000/ws/chat/1/

01:26:09

↓

Saved messages