

Register.php

API

MOONGODB

REDIS

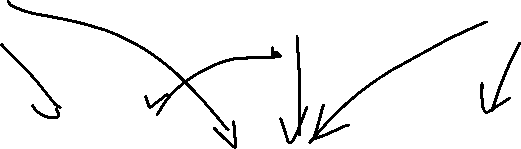
login.php

php.php

Conversation.php

login.html

Register.html



1. Register.html->register.php->S2.py

{username, fname, lname, email, password}

1. S2.py->Redis

Username, email -unique

Password corresponds and it’s not null

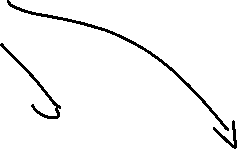
HSET user:username password email fname lastname

1. S2.py->API (POST method)

{username, connected=FALSE, **id**}

API

REDIS

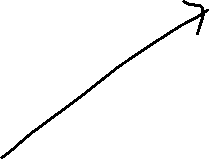


Register.php



Register.html

login.html







1. login.html->login.php->login.py

{username, password}

1. login.py->Redis

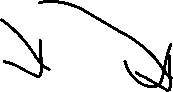
Username +password -match verification

HSET user:username Connection\_hour1 connection\_date1 ... connection\_hour3 connection\_date3

If more than 3 connections => not allowed

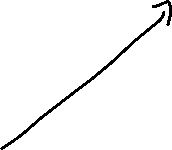
1. S2.py->API (PUT method)

{username, **connected=TRUE**, id}



API

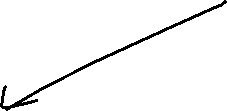
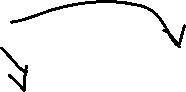
REDIS



login.php

Conversation.php

login.html



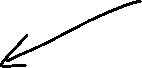
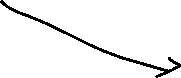
php.php

API

MOONGODB

Existing conversation

New conversation



Conversation.php

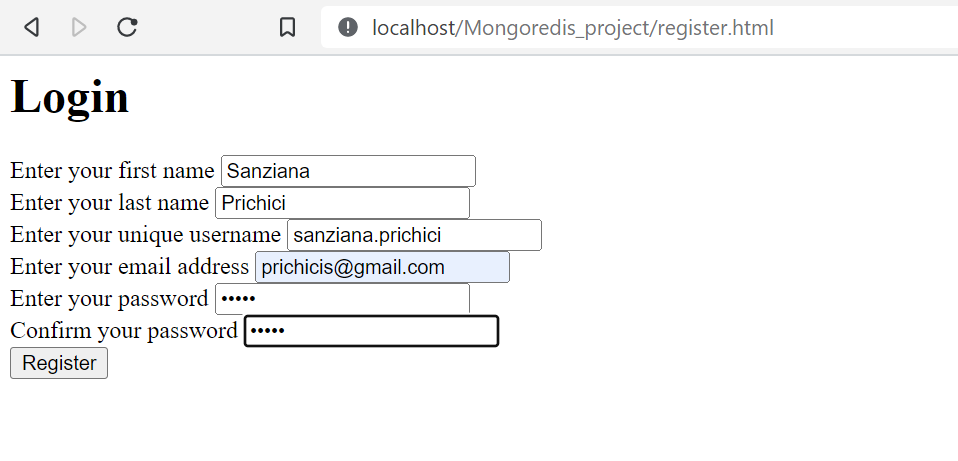
Why API?

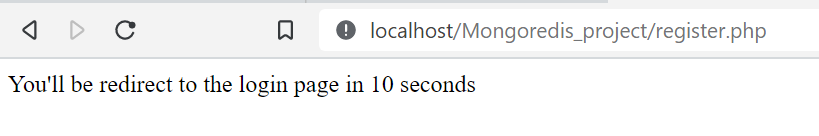
* Descriere componente

Php + python + API

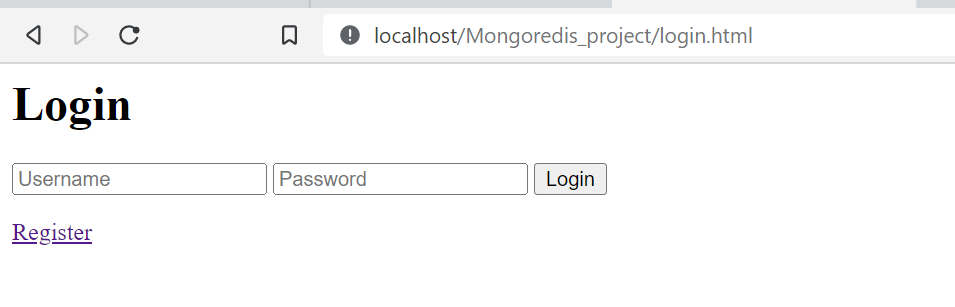
* Descriere register

Php+python+redis+API





* Descriere login

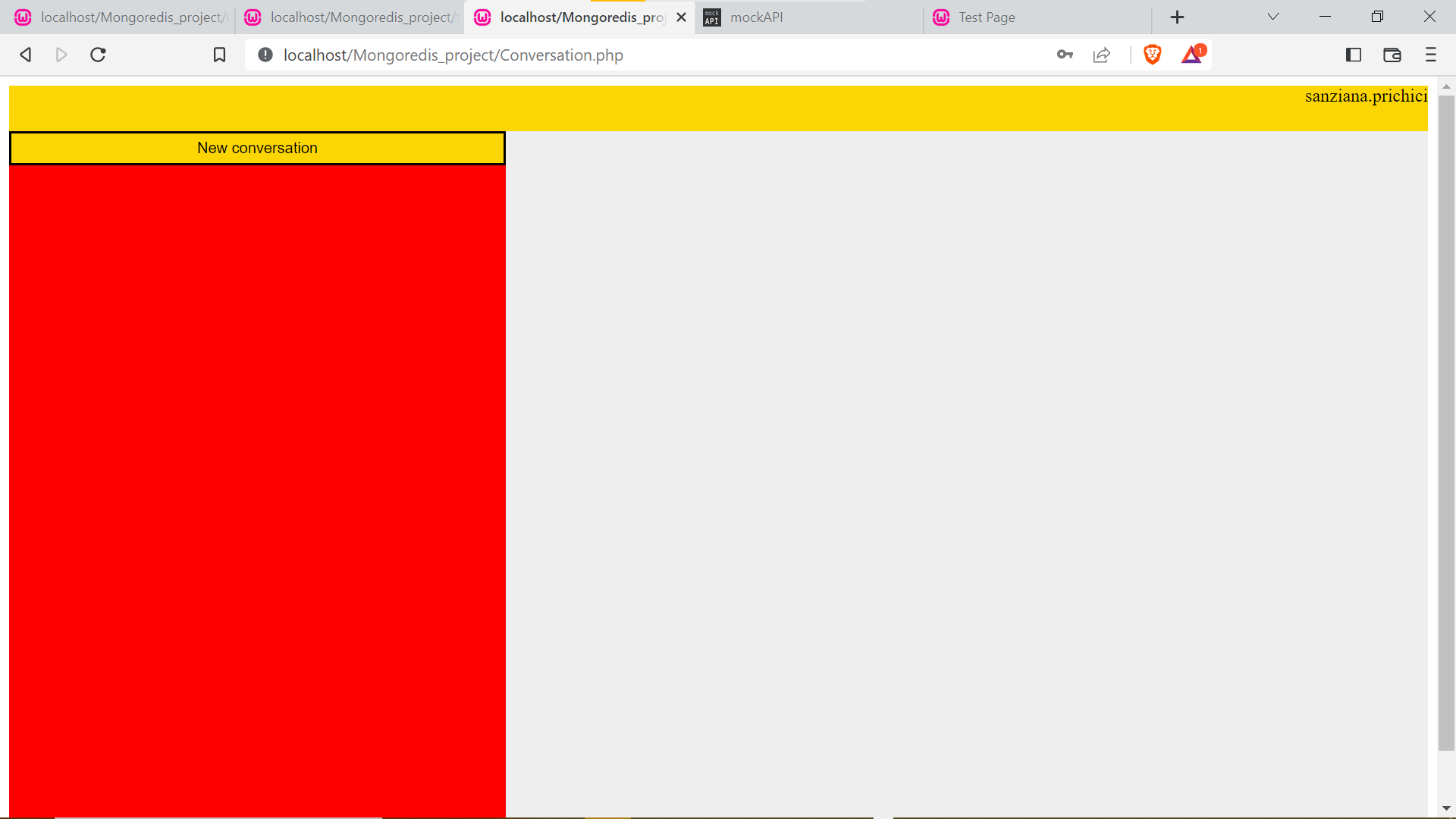


Here, the user can enter his unique username and the password associated to this username. The verification of the relationship between these two will be made by using a python script (S2.py). This script connects to the redis server and

Php+python+redis+API

* Descriere conversation

Php+python+AJAX+Mongo+API



* Mongo side
* Redis side

API -> Client-> PYTHON -> Server ; ->API

$\_SESSION

$\_GET, $\_POST